KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
BADAN PENELITIAN DAN PENGEMBANGAN DAN PERBUKUAN
PUSAT KURIKULUM PERBUKUAN

INFORMATIKA



Mushthofa, dkk.

SMA KELAS X

Disclaimer: Buku ini disiapkan oleh Pemerintah dalam rangka pemenuhan kebutuhan buku pendidikan yang bermutu, murah, dan merata sesuai dengan amanat dalam UU No. 3 Tahun 2017. Buku ini disusun dan ditelaah oleh berbagai pihak di bawah koordinasi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. Buku ini merupakan dokumen hidup yang senantiasa diperbaiki, diperbaharui, dan dimutakhirkan sesuai dengan dinamika kebutuhan dan perubahan zaman. Masukan dari berbagai kalangan yang dialamatkan kepada penulis atau melalui alamat surel buku@kemdikbud.go.id diharapkan dapat meningkatkan kualitas buku ini.

Informatika untuk SMA Kelas X

Penulis
Mushthofa
Wahyono
Auzi Asfarian
Dean Apriana Ramadhan
Hanson Prihantoro Putro
Irya Wisnubhadra
Budiman Saputra
Heni Pratiwi

Penelaah
Inggriani
Paulina Heruningsih Prima Rosa
Adi Mulyanto

Penyelia
Pusat Kurikulum dan Perbukuan

Ilustrator
Malikul Falah

Penata Letak (Desainer)
Nurul Fatonah

Penyunting
Christina Tulalessy
Cecilia Esti Nugraheni

Penerbit
Pusat Kurikulum dan Perbukuan
Badan Penelitian dan Pengembangan dan Perbukuan
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Jalan Gunung Sahari Raya No. 4 Jakarta Pusat

Cetakan pertama, 2021
ISBN 978-062-244-505-0 (no.jil.lengkap)
978-602-244-506-7 (jil. 1)

Isi buku ini menggunakan huruf Newsreader 11/15 pt. (Production Type)
x, 262 hlm.: 17,6 x 25 cm.

Kata Pengantar

Pusat Kurikulum dan Perbukuan, Badan Penelitian dan Pengembangan dan Perbukuan, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi mempunyai tugas penyiapan kebijakan teknis, pelaksanaan, pemantauan, evaluasi, dan pelaporan pelaksanaan pengembangan kurikulum serta pengembangan, pembinaan, dan pengawasan sistem perbukuan. Pada tahun 2020, Pusat Kurikulum dan Perbukuan mengembangkan kurikulum beserta buku teks pelajaran (buku teks utama) yang mengusung semangat merdeka belajar. Adapun kebijakan pengembangan kurikulum ini tertuang dalam Keputusan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 958/P/2020 tentang Capaian Pembelajaran pada Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah.

Kurikulum ini memberikan keleluasan bagi satuan pendidikan dan guru untuk mengembangkan potensinya serta keleluasan bagi siswa untuk belajar sesuai dengan kemampuan dan perkembangannya. Untuk mendukung pelaksanaan Kurikulum tersebut, diperlukan penyediaan buku teks pelajaran yang sesuai dengan kurikulum tersebut. Buku teks pelajaran ini merupakan salah satu bahan pembelajaran bagi siswa dan guru.

Pada tahun 2021, kurikulum dan buku akan diimplementasikan secara terbatas di Sekolah Penggerak. Hal ini sesuai dengan Keputusan Menteri Pendidikan dan Kebudayaan Nomor 1177 Tahun 2020 tentang Program Sekolah Penggerak. Tentunya umpan balik dari guru dan siswa, orang tua, dan masyarakat di Sekolah Penggerak sangat dibutuhkan untuk penyempurnaan kurikulum dan buku teks pelajaran ini.

Selanjutnya, Pusat Kurikulum dan Perbukuan mengucapkan terima kasih kepada semua pihak yang terlibat dalam penyusunan buku ini mulai dari penulis, penelaah, reviewer, supervisor, editor, ilustrator, desainer, dan pihak terkait lainnya yang tidak dapat disebutkan satu per satu. Semoga buku ini dapat bermanfaat untuk meningkatkan mutu pembelajaran.

Jakarta, Juni 2021
Kepala Pusat Kurikulum dan Perbukuan,

Maman Fathurrohman, S.Pd.Si., M.Si., Ph.D.
NIP 19820925 200604 1 001

Prakata

Informatika adalah bidang ilmu mengenai studi perancangan, dan pengembangan sistem komputasi, serta prinsip-prinsip yang menjadi dasar perancangan dan pengembangan tersebut. Informatika dilandasi oleh landasan berpikir yang dinamakan berpikir komputasional (Computational Thinking). Berpikir komputasional ini merupakan suatu kerangka dan proses berpikir yang mencakup perangkat keras, perangkat lunak, dan menalar (reasoning) mengenai sistem dan persoalan. Moda berpikir (thinking mode) ini didukung dan dilengkapi dengan pengetahuan teoretis dan praktis, serta teknik untuk menganalisis, memodelkan dan menyelesaikan persoalan.

Bidang-bidang pengetahuan Informatika dibagi menjadi: Berpikir Komputasional (BK), Teknologi Informasi dan Komunikasi (TIK), Sistem Komputer (SK), Jaringan Komputer dan Internet (JKI), Analisis Data (AD), Algoritma dan Pemrograman (AP), Dampak Sosial Informatika (DSI), dan Praktik Lintas Bidang (PLB). Bidang pengetahuan inilah yang akan dipelajari bersama di buku siswa ini. Isi materi setiap bab terdiri atas konsep/materi/teori terkait bidang tadi dan aktivitas-aktivitas yang dapat dilakukan oleh siswa. Aktivitas ini ada yang individu dan berkelompok. Aktivitas ini juga ada plugged (membutuhkan komputer) dan unplugged (tidak memerlukan komputer). Harapannya siswa dapat memahami konsep dan implementasi Informatika dengan lebih baik dan bermakna.

Buku teks Informatika kelas X telah disesuaikan dengan kebutuhan dari siswa kelas X, yang diharapkan menjadi terampil berpikir, berkarya, dan berteknologi, menjadi *computationally literate*, sebagai anggota masyarakat dan sekaligus warga digital yang berakhlak baik di dunia nyata maupun di dunia digital.

Akhir kata penulis berharap semoga buku siswa ini dapat memberikan manfaat dan dapat digunakan untuk pendamping belajar Informatika sebaik-baiknya. Saran dan kritik membangun sangat penulis harapkan untuk perbaikan penulisan buku lebih lanjut.

Bogor, Februari 2021

Penulis.

Daftar Isi

Kata Pengantar iii
Prakata iv
Daftar Gambar vii
Daftar Tabel x
Bab 1 Informatika dan Keterampilan Generik 1
A. Tentang Informatika di Sekolah Menengah Atas 3
B. Keterampilan Generik 14
Bab 2 Berpikir Komputasional 23
A. Pencarian (*Searching*) 26
B. Pengurutan (*Sorting*) 30
C. Tumpukan (*Stack*) dan Antrean (*Queue*) 37
Bab 3 Teknologi Informasi dan Komunikasi 45
A. Integrasi Konten Aplikasi Perkantoran 47
B. Fitur Lanjut Aplikasi Perkantoran 57
Bab 4 Sistem Komputer 63
A. Komputer dan Komponen Penyusunnya 65
B. Interaksi Manusia dan Komputer 69
C. Kolaborasi dalam Sistem Komputer 70
D. Sistem Operasi 72
Bab 5 Jaringan Komputer dan Internet 91
A. Jaringan Lokal dan Internet 93
B. Komunikasi Data dengan Ponsel 99
C. Proteksi Data Saat Berinternet 104
Bab 6 Analisis Data 109
A. Pengenalan Perkakas Analisis Data 112
B. Koleksi Data 122
C. Visualisasi Data 126
Bab 7 Algoritma dan Pemrograman 135
A. Algoritma 137
B. Bahasa Pemrograman Prosedural 148
C. Bahasa Pemrograman C 149
D. Pengayaan 192
Bab 8 Dampak Sosial Informatika 197
A. Sejarah Perkembangan Komputer 199
B. Aspek Ekonomi dan Hukum dari Produk Informatika 201
C. Informatika untuk Masa Depan 212
D. Karier dan Studi Lanjut di Bidang Informatika 215

Bab 9 Praktika Lintas Bidang Informatika 231
A. Pengarahan Guru dan Observasi 233
B. Pelaksanaan Proyek 236
C. Monitoring dan Evaluasi Proyek 241
Glosarium 245
Daftar Pustaka 250
Sumber Gambar 252
Indeks 254
Profile Penulis 257
Profile Penelaah 260
Profile Editor 261
Profile Ilustrator dan Desainer 262

Daftar Gambar

Gambar 1.1 Peta Konsep Informatika dan Keterampilan Generik 2
Gambar 1.2 Revolusi Industri 4.0 4
Gambar 1.3 Pilar Pengetahuan Informatika 6
Gambar 1.4 Contoh Buku Kerja 13
Gambar 1.5 Pelantikan OSIS (a) Upacara, (b) Nasi Tumpeng untuk acara ramah tamah 16
Gambar 1.6 Contoh grafik (a) Cluster (kelompok) Data, (b) Layer pada aplikasi Geographical Information System, (c) Infografis menjadi pengguna internet anti-hoax... 20
Gambar 1.7 Komik lucu tentang teknologi 20
Gambar 2.1 Peta Konsep Berpikir Komputasional 24
Gambar 2.2 Pencarian (a) buku di perpustakaan, (b) informasi di internet 27
Gambar 2.3 Tombol Back dan Forward pada Firefox 39
Gambar 3.1 Peta Konsep Teknologi Informasi dan Komunikasi 46
Gambar 3.3 *Insert Chart* pada MS Word 49
Gambar 3.2 Muncul Sheet seperti pada Microsoft Excel 49
Gambar 3.4 Muncul Sheet seperti pada Microsoft Excel 50
Gambar 3.5 Ilustrasi *Object Linking* Antardua Dokumen 50
Gambar 3.6 Pembuatan Daftar Isi.. 59
Gambar 4.1 Peta Konsep Sistem Komputer 64
Gambar 4.2 Diagram Blok Konseptual Komputer 65
Gambar 4.3 Contoh Microcomputer 67
Gambar 4.4 Contoh Personal Computer 67
Gambar 4.5 Contoh Mini PC, Coffee Lake-U-based Bean Canyon Intel NUC8i5BEK2 67
Gambar 4.6 Contoh Minicomputer PDP-8e Minicomputer System 68
Gambar 4.7 Salah satu contoh Komputer Mainframe IBM z Systems z13. 68
Gambar 4.8 Sierra/ATS-2 Super Computer 68
Gambar 4.9 Abstraksi Interaksi Perangkat Keras, Perangkat Lunak, dan Pengguna 69
Gambar 4.10 Antarmuka Berbasis Grafis (a) Ikon pada ponsel, (b) Menu, (c) Dialog, textbox, dan button, (d) checkbox dan radio button 69
Gambar 4.11 Contoh Antarmuka Berbasis Perintah (CLI) 70
Gambar 4.12 Valkyrie, Humanoid Buatan NASA 71
Gambar 4.13 Beberapa Contoh Sistem Operasi 72
Gambar 4.14 Alur Sistem Operasi pada Komputer dan "Lapisan Bawang" Sistem Komputer 73
Gambar 4.15 (a) Simulasi Penjadwalan Proses, (b) Simulasi CPU 78
Gambar 4.16 Komponen Internal pada Komputer.. 79
Gambar 4.17 Central Processing Unit 80

Gambar 4.18 Siklus Mesin pada Komputer Sebelum dan Setelah Bekerja 81
Gambar 4.19 Memory Address seperti Kotak Surat 82
Gambar 4.20 Siklus Ambil dan Jalankan (Fetch Execute Cycle) 83
Gambar 5.1 Peta Konsep Jaringan Komputer dan Internet.. 92
Gambar 5.2 Contoh Jaringan Lokal 93
Gambar 5.3 Jaringan Lokal Bisa Mengubah Menjadi Jaringan Internet 94
Gambar 5.4 Diagram Konektivitas Internet Jaringan Berkabel melalui Dial-Up dan ASDL 96
Gambar 5.5 Konektivitas Jaringan LAN 96
Gambar 5.6 Ilustrasi Konektivitas Jaringan Nirkabel via GPRS 97
Gambar 5.7 Ilustrasi Konektivitas Jaringan Nirkabel Wifi 97
Gambar 5.8 Ilustrasi Konektivitas Jaringan Nirkabel melalui Satelit 98
Gambar 5.9 Ilustrasi Studi Kasus Perancangan Jaringan Komputer 99
Gambar 5.10 Salah Satu Alur Kerja Proses Komunikasi Data SMS melalui Ponsel 100
Gambar 5.11 Kabel Data 102
Gambar 5.12 Contoh Ilustrasi Komunikasi Data Menggunakan Bluetooth 102
Gambar 5.13 Contoh Ilustrasi Komunikasi Data Menggunakan Wifi.................... 102
Gambar 5.14 Ilustrasi Berbagai Piranti Elektronik 103
Gambar 5.15 Ilustrasi Jika Pesan di Media Sosial Tidak Dienkripsi 104
Gambar 5.16 Situs kemdikbud.go.id Menggunakan Sertifikat SSL 105
Gambar 6.1 Peta Konsep Analisis Data 110
Gambar 6.2 Contoh Tampilan Halaman Website Pencarian Kerja 111
Gambar 6.3 Tampilan Google Colab Saat Dibuka Pertama Kali............................ 113
Gambar 6.4 Area Kerja Google Colab 114
Gambar 6.5 Contoh sederhana kode program Python 114
Gambar 6.6 Hasil Keluaran Contoh Sederhana Kode Program Python 115
Gambar 6.7 Contoh Assignment dalam Operasi Penjumlahan Sederhana 117
Gambar 6.8 Hasil Keluaran Contoh Sederhana Kode Program Python 118
Gambar 6.9 Contoh Penggunaan for dalam Python 119
Gambar 6.10 Contoh Pengolahan Data yang Bervariasi 120
Gambar 6.11 Contoh Penggunaan try-except dalam Python................................ 121
Gambar 6.12 Contoh sederhana penggunaan import pustaka Pandas 122
Gambar 6.13 Hasil keluaran contoh sederhana penggunaan import pustaka Pandas.. 122
Gambar 6.14 Alur Proses Web Scraping 123
Gambar 6.15 Kode Program untuk Parsing Sebuah Alamat Web 123
Gambar 6.16 Hasil Keluaran dari Kode Program Parsing Website Lowongan Pekerjaan 124
Gambar 6.17 Kode Program untuk Mengoleksi Data Lowongan Pekerjaan......... 125
Gambar 6.18 Hasil Keluaran dari Kode Program Koleksi Data Pekerjaan............ 125
Gambar 6.19 Kode Program untuk Membingkai Data ke dalam Tabel 125
Gambar 6.20 Contoh Hasil Keluaran dari Kode Program yang Menampilkan Tabel 126
Gambar 6.21 Kode Program untuk Preprocessing dalam Menata Data Gaji.. 127

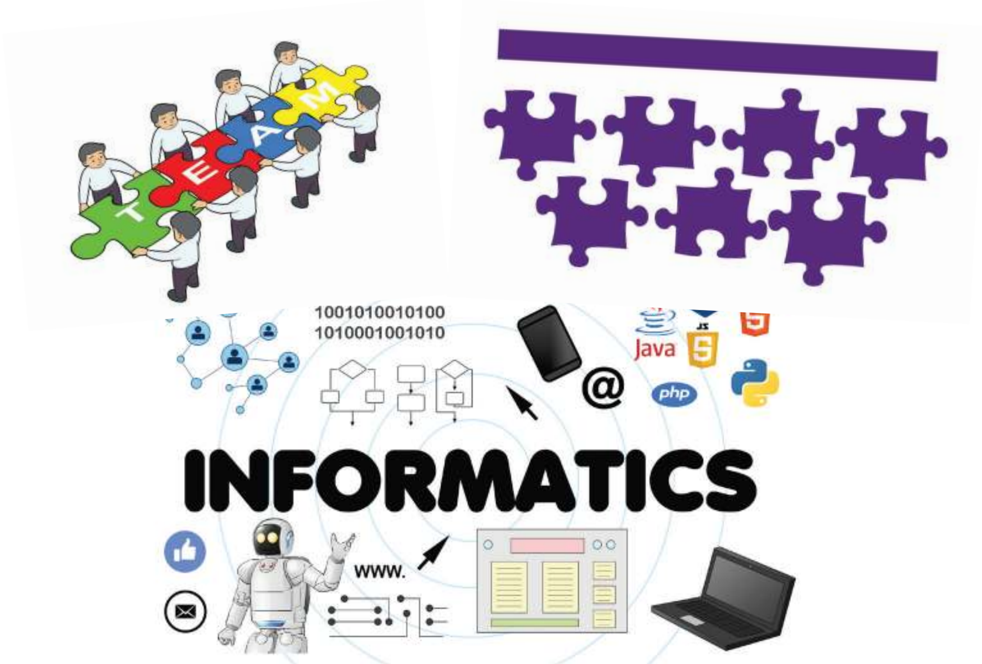
Gambar 6.22 Kode Program untuk Memvisualisasi Data dalam Diagram Batang 128
Gambar 6.23 Contoh Hasil Keluaran dari Kode Program Visualisasi Data 128
Gambar 6.24 Kode Program untuk *Preprocessing* dalam Mengelompokkan Posisi dan Institusi 129
Gambar 6.25 Contoh Hasil Keluaran Setelah Posisi dan Institusi Dikelompokkan 129
Gambar 6.26 Contoh Hasil Keluaran dari Kode Program Visualisasi Data Menggunakan Diagram Lingkaran 130
Gambar 6.27 Contoh Visualisasi Data Secara *Real-Time* 132
Gambar 6.28 Visualisasi untuk Data Berat Badan sebagai Alat Memprediksi Pertumbuhan Ideal Bayi 133
Gambar 7.1 Peta Konsep Algoritma dan Pemrograman 136
Gambar 7.2 Elemen Generik dari Bahasa Pemrograman Prosedural 148
Gambar 7.3 Alur proses membuat program dengan Bahasa C 150
Gambar 7.4 Contoh tampilan sebuah IDE 150
Gambar 7.5 Tampilan *Ideone* 192
Gambar 7.6 Contoh Soal yang Terdapat pada Web TLX yang Dikelola oleh Ikatan Alumni Tim Olimpiade Komputer Indonesia (IA-TOKI) 193
Gambar 8.1 Peta Konsep Dampak Sosial Informatika 198
Gambar 8.2 Electronic Numerical Integrator and Computer (ENIAC), komputer elektronik pertama yang dapat diprogram 199
Gambar 8.3 Ada Lovelace, Pemrogram Komputer Paling Awal di Dunia 200
Gambar 8.4 Beragam bentuk komputer dalam kehidupan sehari-hari (a) Smart Watch (b) Surface Table, (c) Virtual Reality. 200
Gambar 8.5 Ilustrasi Ekonomi Kreatif di Indonesia 203
Gambar 8.6 Komik tentang Seseorang yang Sedang Membeli Komputer 209
Gambar 8.7 Inovasi-Inovasi Unggul di Bidang Informatika Banyak Bermula dari Coretan-Coretan Kreatif Di Buku Catatan atau Jurnal Seseorang. 213
Gambar 8.8 Tujuan Pembangunan Berkelanjutan 213
Gambar 8.9 Ilustrasi Gambaran Dunia di Masa Depan, Masyarakat 5.0 (Society 5.0) (a) abstraksi, (b) contoh aktivitas 214
Gambar 8.10 Ilustrasi Beberapa Profesi di Bidang Informatika 216
Gambar 8.11 Lima Bidang Studi Komputing (Informatika) 225
Gambar 9.1 Peta Konsep Praktik Lintas Bidang 232
Gambar 9.2 Contoh Membuat Desain Aplikasi dengan Menggunakan MS PowerPoint 241

Daftar Tabel

Tabel 4.1 Waktu Kedatangan dan Eksekusi Proses 77
Tabel 7.1 Simbol Diagram Alir Beserta Maknanya 138
Tabel 7.2 Ilustrasi Penelusuran Diagram Alir di Buku Kerja Siswa. 142
Tabel 7.3 Beberapa Tipe Data Pada Bahasa C Beserta Ukuran Memori dan Rentang Nilainya 157
Tabel 7.4 Beberapa Contoh Deklarasi Variabel 158
Tabel 7.5 Spesifikasi Format pada Bahasa C 160
Tabel 7.6 Spesifikasi Program yang Diberikan pada Problem Latihan 163
Tabel 7.7 Operator Matematika pada Bahasa C 166
Tabel 7.8 Operator Logika, Relasional, dan Kesamaan Pada Bahasa C 168
Tabel 7.8 Perbedaan antara Coding dan Pemrograman 191
Tabel 7.9 Padanan Sintaks Bahasa C dan Bahasa Python 195
Tabel 8.1 Profesi di Bidang Informatika 218
Tabel 8.2 Jenis-Jenis Profesi di Bidang Informatika 219

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Irya Wisnubhadra
ISBN : 978-602-244-506-7

Bab 1
Informatika dan Keterampilan Generik



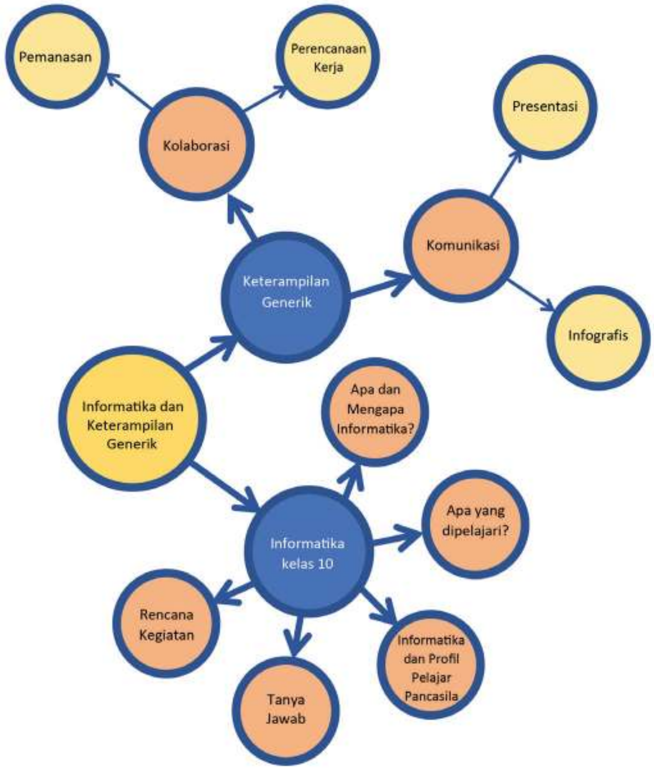
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu berkolaborasi dalam tim dan mengomunikasikan hasilnya dengan efektif dan efisien.

Pertanyaan Pemantik

Kapan terakhir kali kalian bekerja dalam kelompok dan presentasi? Apakah kalian melakukan kesepakatan sebelum mulai mengerjakan tugas? Apakah lebih menyenangkan dibandingkan jika kerja sendiri? Apakah presentasi kalian menarik?

Peta Konsep



Gambar 1.1 Peta Konsep Informatika dan Keterampilan Generik

Apersepsi

Kalian mungkin banyak yang pernah menggunakan aplikasi permainan pada telepon pintar? Aplikasi permainan adalah salah satu produk yang dihasilkan berkat ilmu Informatika. Pengembangan produk yang kompleks perlu kolaborasi tim yang baik agar efektif dan hasilnya sesuai dengan keinginan.

Kata Kunci

Informatika, bidang pengetahuan, keterampilan generik, bekerja dalam kelompok, perencanaan kerja, pengomunikasian hasil kerja, presentasi, infografis

A. Tentang Informatika di Sekolah Menengah Atas

Informatika adalah mata pelajaran yang baru diterbitkan buku pelajaran resminya. Informasi berikut adalah penjelasan secara ringkas tentang kurikulum Informatika.

1. Apa dan Mengapa Informatika?

Informatika adalah bidang ilmu mengenai studi, perancangan, dan pembuatan sistem komputasi, serta prinsip-prinsip yang menjadi dasar perancangan tersebut. Komputasi adalah ilmu yang berkaitan dengan pemodelan matematika dan penggunaan komputer untuk memecahkan masalah-masalah sains. Istilah *informatika* sepadan dengan istilah dalam bahasa Inggris *Informatics, Computing*, atau *Computer Science*. Informatika mencakup pemodelan dari "komputasi" dan aplikasinya dalam pengembangan sistem komputer. Apa itu komputasi? Menurut KBBI, komputasi adalah: (1) penghitungan dengan menggunakan komputer; (2) dalam matematika, penghitungan dengan menggunakan bilangan-bilangan atau peubah-peubah yang dilaksanakan berdasarkan urutan langkah yang diberikan.

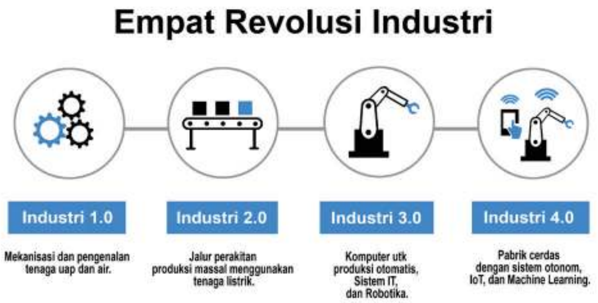
Landasan berpikir untuk belajar informatika dinamakan berpikir komputasional (*Computational Thinking*). Berpikir komputasional ini merupakan suatu kerangka dan proses berpikir yang mencakup perangkat keras, perangkat lunak, dan menalar (*reasoning*) mengenai sistem dan persoalan. Moda berpikir (*thinking mode*) ini didukung dan dilengkapi dengan pengetahuan teoritis dan praktis, serta teknik untuk menganalisis, memodelkan dan memecahkan persoalan. Siswa yang belajar informatika akan mendalami bagaimana suatu "sistem komputasional" berfungsi, baik yang mengandung komputer maupun tidak.

Setelah melalui perkembangan lebih dari 20 tahun, informatika telah menjadi salah satu disiplin ilmu yang saat ini sudah berdiri sendiri. IInformatika dapat dipandang sebagai sebuah cabang ilmu yang tersendiri karena membawa seseorang ke suatu cara berpikir yang unik (*computational thinking*), dan berbeda dari bidang ilmu lainnya. Ide-idenya tahan lama (20 tahun lebih dan masih terpakai sampai sekarang), dan prinsip-prinsip intinya dapat diajarkan tanpa bergantung kepada teknologi tertentu.

Semula, informatika hanya diajarkan di tingkat Perguruan Tinggi. Sekarang, di berbagai negara di dunia, termasuk Indonesia, informatika sudah mulai diajarkan di tingkat pendidikan dini, dasar dan menengah.

Informatika di Era Industri 4.0

Saat ini manusia hidup di era digital yang sering disebut Industri 4.0. Dalam industri 4.0, banyak hal dilakukan oleh manusia dengan memanfaatkan mesin cerdas berbasis komputer dan internet. Masyarakat sekarang ini



Gambar 1.2 Revolusi Industri 4.0
sumber: Dokumen Kemendikbud, 2021 di adaptasi dari https://www.spectralengines.com/articles/industry-4-0-and-how-smart-sensors-make-the-difference

disebut masyarakat 5.0 (*society 5.0*) karena hidup di dunia fisik sekaligus di dunia siber (maya) saat melakukan kegiatan daring. Banyak kegiatan dilakukan manusia secara fisik maupun secara daring dengan pemanfaatan teknologi komputer dan internet. Kalian tentu bisa mengingat pengalaman kegiatan kalian yang didukung teknologi komputer dan internet, mulai dari berkomunikasi dengan media sosial, bermain *game online*, memesan makanan, memesan transportasi, mendaftar sekolah, belajar secara daring, dan sebagainya.

Dunia yang seperti ini menjadi makin cepat berubah, banyak hal menjadi tidak pasti, persoalan yang dihadapi manusia makin rumit, dan beragam cara untuk melakukan hal yang sama. Sebagai contoh, dulu orang harus pergi ke toko atau ke pasar untuk berbelanja. Penjual juga harus mempunyai bangunan kios atau toko. Pemilik toko mempekerjakan pegawai untuk mengurus gudang, menjadi kasir, dan menangani berbagai pekerjaan lain. Pembeli juga harus membawa uang tunai untuk membayar. Sekarang, orang bisa berbelanja dari rumah, tidak harus pergi ke mana-mana walaupun tidak ada penjual keliling yang datang. Penjual juga bisa berjualan dari rumah tanpa perlu memiliki bangunan kios atau toko. Pembeli tidak harus mengambil uang di bank untuk membayar barang yang dibelinya karena bisa membayar dengan menggunakan uang digital. Contoh lain yang telah kita alami sendiri ialah pembelajaran daring (dalam jaringan) selama masa pandemi. Di dunia industri, lebih banyak lagi hal yang kini dilakukan dengan bantuan teknologi komputer. Segala sesuatu diselesaikan secara otomatis karena sudah ada robot-robot yang mampu mengerjakan pekerjaan rutin yang berulang-ulang. Cara orang melakukan kegiatan menjadi berubah karena teknologi komputer dan internet yang dimanfaatkan di berbagai bidang.

Nah, situasi seperti contoh di atas memudahkan orang untuk melakukan kegiatan berbelanja, belajar, dan sebagainya. Namun coba pikirkan, bagaimana nasib para pekerja yang semula bekerja sebagai kasir, pegawai gudang, pegawai yang melakukan pengepakan barang di pabrik, *teller* di bank, dan sebagainya? Ada banyak orang yang kehilangan pekerjaan karena teknologi

mengambil alih hal-hal yang biasa mereka kerjakan. Di masa depan, pasti akan lebih banyak lagi peran teknologi dalam kegiatan manusia. Mungkin juga akan makin banyak jenis pekerjaan yang hilang. Sebaliknya, pencipta teknologi itulah yang makin dibutuhkan.

Menghadapi situasi tersebut, siswa yang sedang belajar di jenjang SMA perlu berpikir kritis dan mulai memikirkan masa depan. Tentu, tidak semua orang akan berkarir di bidang komputer. Zaman sekarang, kesempatan untuk berkarya di berbagai bidang sangat terbuka. Namun, hampir semua ilmu lain membutuhkan komputasi atau penyimpanan informasi serta pembentukan pengetahuan yang membutuhkan komputer. Bidang-bidang yang semula sulit terjangkau untuk diriset, makin tertolong dengan kemampuan komputasi, misalnya untuk menyimulasikan alam raya dan benda langit, atau menengok struktur mikro sel hidup atau material seperti molekul, atom, atau gen manusia. Semua profesi mulai ahli sejarah, ahli ekonomi, guru, apoteker, dokter, perawat, ilmuwan, insinyur, pedagang, semua membutuhkan alat bantu kerja yang "mengandung" komputer. Bahkan, di rumah pun, untuk sekadar berkomunikasi, ponsel pintar sudah dilengkapi dengan aplikasi. Produk hiburan seperti permainan, atau video yang dinikmati jutaan pengguna lewat kanal siaran, bahkan film drama adalah karya digital yang diciptakan para produsen dengan bantuan ahli digital media dan juga ahli seni.

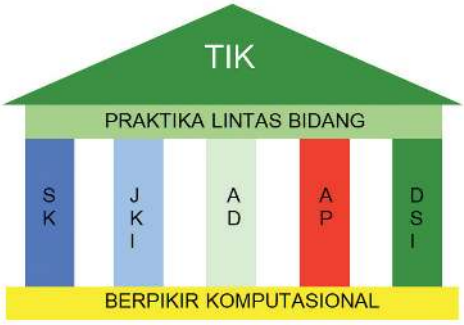
Nah, untuk itulah siswa, SMA akan belajar mata pelajaran informatika ini. Selain belajar berpikir komputasional dalam menghadapi persoalan sehari-hari serta cara memanfaatkan teknologi komunikasi lewat jaringan, perangkat keras dan perangkat lunak dengan lebih efisien, siswa SMA perlu mempelajari teori-teori informatika dengan lebih mendalam, terutama tentang analisis data, pemrograman, dan dampak informatika dalam kehidupan dan pekerjaan. Teori saja tentu tidak cukup sebab baru berguna jika dapat dipraktikkan.

Dengan demikian, siswa SMA akan bisa menyiapkan diri untuk menghadapi masa depan yang pasti akan makin cepat kemajuan teknologinya, dan membutuhkan komputer dan komputasi di bidang apa pun. Melalui mata pelajaran informatika, diharapkan di masa mendatang, pelajar Indonesia tidak hanya mampu mempergunakan teknologi komputer, tetapi juga memahami strategi pemecahan persoalan dan teknologi komputasi tepat guna yang berpotensi untuk dimanfaatkan dalam menyelesaikan persoalan. Setelah lulus SMA, bahkan juga akan tahu dan mampu bagaimana mengembangkan program dan sistem komputasi sederhana untuk mengatasi berbagai persoalan. Dengan demikian, akan makin banyak orang Indonesia yang mampu menciptakan teknologi berbasis komputer yang bermanfaat bagi banyak orang. Menarik, bukan?

2. Apa yang akan Dipelajari dalam Mata Pelajaran Informatika?

Informatika mencakup aspek teoretis dan praktis, mendorong kalian untuk mengembangkan daya pikir kritis dan kreatif untuk menghasilkan penemuan (*invention*) terkait komputer dan sistem komputasi. Kalian diharapkan mampu mengaplikasikan prinsip-prinsip ilmiah yang dipelajarinya untuk memahami dunia nyata, kemudian menciptakan perangkat keras dan perangkat lunak yang memudahkan kehidupan. Dengan mengombinasikan dari prinsip, praktik dan penemuan (*principles, practice, and invention*), kalian akan menjadi kreatif dan merasakan manfaatnya. Karena tidak hanya teoritis, penerapan informatika akan memberikan "*surprise*", kejutan karena produknya menghasilkan sesuatu yang dapat diamati dan berfungsi dengan baik dan dapat digunakan, bahkan dapat memunculkan keindahan ("*that is so beautiful*").

Kalian akan mempelajari bidang-bidang pengetahuan Informatika, yang dikelompokkan menjadi Berpikir Komputasional (BK), Teknologi Informasi dan Komunikasi (TIK), Sistem Komputer (SK), Jaringan Komputer dan Internet (JKI), Analisis Data (AD), Algoritma dan Pemrograman (AP), Dampak Sosial Informatika (DSI), dan Praktik Lintas Bidang (PLB). Dalam bahasa Inggris, bidang itu disebut: Computational Thinking (CT), Information and Communication Technology (ICT), Computing System (CE), Computer Network (NW), Data Analysis (DA), Algorithm & Programming (AP), Social Impact of Informatics (SOC), Computing Practices (CP). Gambar 1.3 berikut ini menggambarkan bidang-bidang pengetahuan atau elemen pengetahuan informatika tersebut.



Gambar 1.3 Pilar Pengetahuan Informatika
Sumber: Dokumen Kemendikbud, 2021

Keterangan:

TIK: Teknologi Informasi dan Komunikasi
SK: Sistem Komputer
JKI: Jaringan Komputer/Internet
AD: Analisis Data
AP: Algoritma dan Pemrograman
DSI: Dampak Sosial Informatika
PLB: Praktik Lintas Bidang
BK: Berpikir Komputasional

a. Apa yang dipelajari di kelas X? Apa bedanya dengan Informatika SMP?

Materi yang disajikan dalam buku siswa ini dirancang untuk siswa yang sudah tamat belajar materi Informatika di fase SMP. Setelah belajar informatika selama 3 tahun selama SMP, tentunya siswa kelas X tidak asing lagi dengan dasar-dasar informatika. Kini saatnya siswa kelas X berlatih untuk berpikir lebih kritis dan lebih kreatif, serta belajar hal yang lebih konseptual dan praktik yang lebih mendalam dengan menggunakan teknologi.

Berikut ini adalah gambaran umum materi pelajaran Informatika bagi kelas X:

1. Berpikir Komputasional: siswa kelas X akan lebih fokus ke persoalan yang solusinya membutuhkan program komputer, sehingga perlu belajar tentang strategi penyelesaian persoalan yang tidak intuitif dan naif, tetapi yang didasari bagaimana komputasi akan dilakukan dengan lebih efisien dan optimal.
2. Teknologi Informasi dan Komunikasi: dalam bagian ini siswa kelas X akan memahami integrasi antar aplikasi, memanfaatkan fitur aplikasi yang lebih advanced (lanjut) dibandingkan fase sebelumnya, dan memanfaatkan fitur otomasi yang tersedia.
3. Sistem Komputer: siswa kelas X akan belajar mengenai perangkat keras komputer dan menengok lebih dalam serta lebih rinci ke dalam setiap komponen komputer dan sistem operasi. Siswa kelas X akan mempelajari lebih rinci bagaimana data dan program disimpan dalam memori, dan bagaimana instruksi diproses.
4. Analisis Data: dalam bagian ini siswa kelas X akan belajar tentang pengelolaan dan penyajian data bervolume lebih besar dan melakukan analisis yang lebih kompleks dari fase sebelumnya. Siswa kelas X juga akan belajar bagaimana mengotomasi pengambilan data dari internet. Ingat, bahwa setiap detik, di internet muncul informasi baru. Kita tidak mungkin mengejar pengambilan data jika tidak melakukan otomasi
5. Algoritma dan Pemrograman: siswa kelas X akan belajar membuat program dengan menggunakan bahasa pemrograman tekstual, dengan menerapkan strategi berpikir komputasional yang dikembangkan pada materi Berpikir Komputasional. Peralihan dari program visual ke program tekstual membutuhkan abstraksi yang lebih tinggi dan lebih sulit, tetapi ini menantang!
6. Dampak Sosial Informatika: dalam bagian ini siswa kelas X akan belajar mengenai dampak informatika di berbagai bidang seperti ekonomi, hukum dan keterbukaan informasi. Siswa kelas X juga akan mendapatkan gambaran dampak informatika di berbagai bidang serta karir apa saja yang terbuka di bidang informatika.

7. Praktik Lintas Bidang : selain mempelajari materi-materi di atas, siswa kelas X juga akan belajar bekerjasama secara kolaboratif untuk membuat rancangan solusi persoalan sosial yang harus mengandung perangkat lunak aplikasi.

3. Informatika dan Profil Pelajar Pancasila

Untuk menyiapkan diri menghadapi masa depan di era digital, siswa kelas X juga perlu membangun kebiasaan-kebiasaan baik yang menumbuhkan karakter yang baik di dunia nyata maupun dunia digital, karena citra bangsa Indonesia terpapar di dunia digital yang sangat terbuka akibat kemunculan dan posting konten orang Indonesia. Bagi pelajar Indonesia, karakter yang baik ini dirumuskan sebagai profil pelajar Pancasila, yang memiliki ciri-ciri berikut ini: (a) Beriman, bertakwa kepada Tuhan YME, dan berakhlak mulia, (b) Berkebhinekaan global, (c) Mandiri, (d) Bergotong Royong, (e) Bernalar kritis, dan (f) Kreatif.

Nah, melalui mata pelajaran informatika, kalian akan belajar juga menumbuhkan karakter-karakter tersebut, sebagai warga dunia nyata dan sebagai warga dunia maya. Semua elemen pembelajaran pada buku ini diharapkan dapat mengasah kemampuan kalian bernalar kritis dan kreatif, serta melatih kalian mengaitkan satu elemen pengetahuan dengan pengetahuan lainnya.

Dalam proses belajar berbagai materi dalam informatika, kalian akan belajar untuk mandiri saat mengerjakan tugas-tugas yang diberikan oleh guru. Kalian juga akan mengembangkan kemampuan bergotong royong, bernalar kritis dan kreatif saat berdiskusi dan berkreasi memecahkan masalah atau membuat proyek bersama teman-teman kalian. Teman-teman kalian mungkin berasal dari berbagai daerah, bahkan mungkin kalian akan bertemu dengan teman-teman dari berbagai negara melalui dunia maya. Saat itulah kalian akan belajar menumbuhkan toleransi dan saling menghargai antar teman yang beraneka ragam latar belakangnya. Hal itu tentu dilakukan karena kita meyakini bahwa semua makhluk ciptaan Tuhan YME itu baik dan harus dihormati. Cara berkomunikasi yang baik dengan sesama menunjukkan akhlak yang mulia.

4. Tanya Jawab

Teknologi Informasi dan Komunikasi (TIK) berbeda dengan Informatika. Informatika berbeda dengan Teknologi Informasi. Oleh karena itu, kalian harus menggunakan kedua istilah tersebut dengan tepat dan sesuai konteksnya. Teknologi Informasi, Teknologi Informasi dan Komunikasi, dan Informatika sudah dibahas dan didalami semasa Fase SMP. Di bagian ini, beberapa hal yang lebih mendalam tentang informatika yang sering menjadi pertanyaan banyak orang, dan mungkin juga pertanyaan siswa kelas X akan disajikan beserta jawabannya.

Pertanyaan : Apa persamaan dan perbedaan informatika fase SMP dengan Fase SMA?

Jawaban : Elemen mata pelajaran informatika sama, tetapi pembahasannya akan lebih mendalam dan praktiknya lebih banyak menggunakan komputer dan perkakas yang sesuai dengan bidang yang dipelajari.

Proyek yang dikerjakan lebih luas cakupannya, dan menuntut siswa menggali permasalahan secara lebih mendalam, untuk merancang dan menghasilkan solusi yang efektif dan efisien

Pertanyaan : Problem Solving? Jenis apa yang akan dilatih untuk diselesaikan pada Fase SMA?

Jika dalam fase sebelumnya persoalan lebih dikaitkan dengan persoalan sehari-hari, pada Fase SMA kita akan membahas kasus-kasus yang lebih bernuansa informatika, artinya yang solusinya lebih mengarah ke solusi yang terprogram (bisa dijalankan oleh komputer). Program komputer yang dibuat semasa fase SMP lebih sebagai program kreativitas, sedangkan program komputer yang akan dikembangkan di kelas X akan berupa program komputer yang akan berguna untuk mendukung dalam memodelkan komputasi.

Pertanyaan : Apakah Informatika perlu untuk bidang sosial dan humanitas (soshum) dan seni?

Jawaban : Bidang sosial dan humanitas serta seni juga memerlukan komputer. Kalaupun siswa tidak akan memilih karir menjadi pengembang sistem komputasi, pengolahan data dalam suatu riset sosial akan membutuhkan perkakas pengolahan yang pasti membutuhkan informatika. Perkakas informatika ini disederhanakan antarmukanya sehingga pengguna bidang non informatika hanya perlu memandang perkakas informatika sebagai black box. Artinya, pengguna bidang informatika tidak perlu tahu detailnya, tinggal memakai seperti banyak pemakai televisi atau kulkas tanpa perlu mengerti rangkaian elektronika di dalamnya. Akan tetapi, memahami struktur dari perkakas dengan baik dan memahami logika pengembangannya akan membantu pengembang perkakas. Sistem sosial banyak dimodelkan dan disimulasikan, untuk mengkaitkan masa lalu, masa kini, dan masa depan. Simulasi memungkinkan kita melakukan percobaan dengan mengubah-ubah parameter, dan melakukan “what if analysis” atau “Bagaimana jika.... “

Pertanyaan : Apa hubungan informatika dengan matematika?

Jawaban : Matematika adalah akar dari informatika, terutama untuk bidang logika dan matematika diskrit, statistika, dan analisis data. Informatika memadukan teori-teori matematika dan bidang lain dengan prinsip dan teori informatika, untuk menghasilkan model komputasi yang dapat disimulasikan dengan komputer.

Pertanyaan : Apa hubungan informatika dengan ilmu pengetahuan alam?

Jawaban : Informatika akan menjadi alat berpikir saat mengembangkan model-model komputasi untuk bidang kimia, biologi, fisika, atau lainnya. Kemajuan

sains banyak dicapai dengan kemajuan bidang informatika. Sebaliknya, kebutuhan pemodelan dan komputasi sains juga menyebabkan teknik untuk pengolahan, analisis dan interpretasi data bagi sains semakin baik.

Pertanyaan : Apa hubungan informatika dengan engineering (rekayasa)?

Jawaban : Dalam proses rekayasa, insinyur insinyur menentukan kebutuhan, melakukan analisis, perancangan, implementasi dan pengujian produk/artefak. Insinyur sipil akan menghasilkan berupa bangunan (gedung, jembatan, atau lainnya). Insinyur mesin menghasilkan mesin dan engine untuk berbagai alat mulai engine pesawat, mobil, atau alat rumah tangga sederhana. Praktik lintas bidang informatika memberi kesempatan siswa untuk melakukan praktik proses rekayasa, yang meliputi analisis, perancangan, implementasi dan pengujian sebelum mengembangkan produk/artefak komputasional.

Pertanyaan : Kegiatan apa saja yang akan dilakukan dalam pelajaran informatika? Apakah sama dengan SMP?

Jawaban : Ya, masih sama dengan saat fase SMP, siswa kelas X akan mengalami pembelajaran berbasis proyek (*Project Based Learning*/PBL), pembelajaran berpusat pada siswa (*Student Centered Learning*/SCL), dan pembelajaran berbasis inkuiri (*Inquiry Based Learning*/IBL).

Dalam pembelajaran berbasis proyek, siswa secara berkelompok akan membuat karya (proyek) untuk menerapkan pengetahuan yang telah dipelajari.

Dalam pembelajaran berbasis inkuiri, siswa akan melakukan eksplorasi secara mandiri untuk suatu topik yang diberikan oleh guru untuk merumuskan pertanyaan-pertanyaan atau persoalan dari permasalahan yang ada.

Dalam PBL dan IBL, siswa adalah aktor utama di kelas karena diberi kepercayaan untuk belajar menjadi mandiri, menemukan ide, berpendapat, berkreasi, dan menyampaikannya ke orang lain. Pembelajaran seperti inilah yang disebut sebagai pembelajaran yang berpusat pada siswa. Proses ini akan menumbuhkembangkan siswa menjadi ilmuwan yang berpikir kritis, kreatif, dan inovatif dalam mengajukan solusi-solusi.

Pertanyaan : Apa yang dimaksud dengan transformasi digital?

Jawaban : Transformasi digital mengacu pada proses dan strategi menggunakan teknologi digital untuk mengubah praktik pekerjaan dan bahkan kehidupan sehari-hari menjadi "digital" yaitu lebih banyak memakai sistem komputer. Hal ini semakin mungkin karena masyarakatnya juga semakin "melek digital" (digital literate), perangkat keras semakin canggih dan justru menjadi murah serta berkinerja tinggi (powerful), perangkat lunak semakin mudah dioperasikan, dan teknologi juga memungkinkan sistem beroperasi lebih efisien dan memberikan nilai manfaat bagi kehidupan.

Pertanyaan : Apa itu big data?
Jawaban : Istilah big data banyak disebut-sebut. Big data adalah istilah yang menggambarkan data terstruktur maupun tidak terstruktur yang bervolume besar, yang cepat pertumbuhannya. Data tersebut membanjiri dunia digital karena semakin banyak orang mengisi konten lewat aplikasi, misalnya berkomunikasi/bersosialisasi melalui media sosial. Namun bukan jumlah data yang penting. Apa yang dilakukan organisasi dengan data itulah yang penting. Big data dapat dianalisis untuk selanjutnya dipakai mengarahkan keputusan seseorang, memberikan rekomendasi, atau lainnya. Dalam organisasi dan bisnis, big data dapat dipakai untuk membantu proses pengambilan keputusan yang strategis dengan menggali halaman web yang dapat diakses secara publik.

Pertanyaan : Apa itu Artificial Intelligence?
Semula, kecerdasan buatan hanya dikaitkan dengan robot yang menggantikan peran manusia. Sekarang, kecerdasan buatan tidak hanya berkaitan dengan robot. Kecerdasan buatan (*Artificial Intelligence*/AI) memungkinkan mesin untuk belajar dari pengalaman, menyesuaikan input-input baru dan melaksanakan tugas seperti manusia. Sebagian besar contoh AI yang kita dengar dewasa ini – mulai dari komputer yang bermain catur hingga mobil tanpa awak (*autonomous car*) – sangat mengandalkan pembelajaran mendalam dan pemrosesan bahasa alami. Dengan menggunakan teknologi ini, komputer dapat dilatih untuk menyelesaikan tugas-tugas tertentu dengan memproses sejumlah besar data dan mengenali pola dalam data. Nah, jika mesin (dapat berupa perangkat lunak saja, atau gabungan perangkat lunak dan perangkat keras) dapat belajar, tentunya Siswa SMA kelas X juga bisa belajar jauh lebih baik dari mesin kan? Manusia harus unggul dari mesin, karena manusia menciptakan mesin!

5. Rencana Kegiatan

Selama dua semester, kalian akan melaksanakan serangkaian kegiatan, yang akan disampaikan oleh guru. Kegiatan tersebut urutannya akan ditentukan oleh guru, dan bukan belum tentu oleh urutan dalam buku ini, karena bahan dalam buku ini memang sengaja dibuat lebih banyak, dan tidak semuanya akan dibahas di kelas. Buku ini dapat lebih mudah dipahami oleh siswa yang sudah tamat belajar buku Informatika kelas 7, 8 dan 9.

Sebelum belajar tentang informatika, ada beberapa kemampuan umum yang akan dipakai sepanjang pembelajaran, yang penting untuk kalian kenal dan alami lebih dulu, yaitu bekerja dalam kelompok dan bagaimana mengkomunikasikan hasil kerja dengan baik. Dua hal ini akan disimulasikan di awal pekerjaan, dan diharapkan dapat kalian praktikkan dengan semakin baik seiring dengan berjalannya waktu.

Catatlah urutan kegiatan yang direncanakan oleh guru menjadi dua buah tabel Program Kegiatan Semester. Tuliskan setiap tabel dalam satu lembar kertas, dan masukkan dalam Map Buku Kerjamu.

6. Rekaman Kegiatan

Setiap kali melakukan kegiatan, kalian harus mencatat kegiatan dalam jurnal, dan juga membuat catatan, mengerjakan latihan, mengisi Lembar Kerja Siswa yang dibagikan.

Siapkanlah jurnal siswa sebelum memulai mata pelajaran Informatika. Ini adalah contoh dari Jurnal

a. Jurnal Siswa

Nama : ....
Kelas/Rombel : ....
Semester : 1 / 2, Tahun Ajaran .... Mulai Tanggal ... s.d. ....

Minggu ke	Aktivitas	Topik yang Kupelajari	Rangkuman Refleksiku
1			
2			
...			

b. Evaluasi diri dan Refleksiku di akhir semester:

1. Aku sangat senang/senang/kurang senang/tidak senang
   Karena ....
2. Aku merasa sudah belajar dengan baik dan berusaha, jika dinilai skala 1 s.d. 4, aku akan menilai usahaku ...
3. Catatanku: ....

c. Buku Kerja Siswa

Buku Kerja Siswa adalah himpunan semua hasil belajar kalian selama satu tahun. Dengan menyusun buku kerja siswa sebagai catatan terpisah, kalian sudah menyumbang ke lingkungan (*green environment*), karena Buku Siswa dapat dipakai oleh adik-adikmu. Oleh karena itu, tolong jangan mencorat-coret Buku Siswa. Jika kalian perlu membuat catatan, ambil selembar kertas dan tuliskan catatannya dikaitkan dengan nomor halaman Buku Siswa.

Wujudkan Buku kerjamu dengan menggunakan kertas binder, yaitu kertas lepasan yang dibundel dalam map, dapat disisipkan sesuai keperluan. Dengan menggunakan map dan kertas binder, kalian berlatih untuk menerapkan *Computational Thinking*, mengorganisasi artefak hasil tugas dan hasil belajarmu dengan rapi, terstruktur sehingga dengan mudah dapat dicari kembali. Setiap lembar kertas kerja harus mengandung identitas namamu,

topik yang dipelajari dan nomor halaman. Nomor halaman hanya perlu urut dalam satu kelompok laporan. Rencanakanlah dengan baik penomoran halaman buku kerjamu.

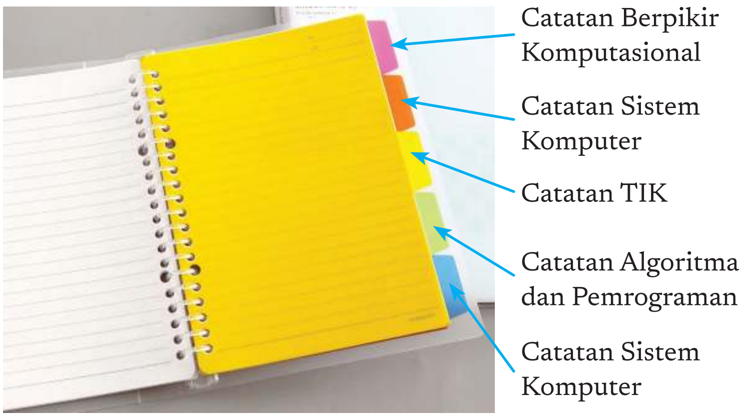
Ini Informatika! Menyusun kode lembar kerja, menyimpan sebagai arsip, dan dengan mudah kalian dapat menemukan kembali saat diperlukan. Hal yang sama akan kalian lakukan untuk file dan folder yang akan kalian simpan dalam komputer, jika kalian mempunyai komputer/laptop sendiri. Itu hanya masalah media. Kerapian kalian menyusun lembar kerja siswa, mencerminkan kemampuanmu mengorganisasi file dan folder di komputer!

Dengan mengorganisasi lembar kerjamu sebagai kertas binder*,* kalian juga dapat mengurutkan sesuai dengan urutan yang diperlukan. Saat suatu tugas selesai dikerjakan, kertas terlepas dapat diperiksa oleh guru, kalian tetap dapat mengacu ke semua bahan yang ada dalam map. Jangan lupa mengarsip saat lembar sudah dikembalikan oleh gurumu. Catat di halaman akhir lembar tugas, kapan tugas kalian serahkan, dan kapan dikembalikan oleh gurumu.

Kreativitasmu sangat dihargai dan harus dipupuk. Kamu boleh menggambar atau menambahkan ilustrasi pada setiap catatan, jika masih mempunyai waktu. Sisipkan lembar pemisah di antara kelompok berkas untuk memudahkan kalian mengakses suatu lembar kerja tertentu dengan lebih cepat. Jika tersedia komputer dan *printer*, kalian juga boleh mencetak dan mengarsip cetakan komputer menjadi bagian bukumu. Buku ini akan menjadi buku kenang-kenangan (memori) belajar yang menyenangkan.

Caramu memelihara buku kerja, dan kerapianmu dalam mengorganisasi isinya, menunjukkan kemampuanmu dalam mengorganisasi informasi dalam komputer. Ini suatu praktik Informatika! Oleh sebab itu, buku kerjamu di akhir tahun akan dinilai oleh secara menyeluruh.

Berikut adalah contoh buku kerja yang diharapkan



Gambar 1.4 Contoh Buku Kerja

B. Keterampilan Generik

Dalam konsep kurikulum informatika, dikenal adanya 7 aspek praktik lintas bidang , di mana 2 di antaranya bersifat umum dan akan dipraktikkan di mana-mana. Tentunya, tidak semuanya harus dipraktikkan sekaligus pada satu aktivitas.

Tujuh aspek praktika informatika itu adalah:

1. Membina dan menumbuhkan budaya kerja masyarakat digital dalam tim yang inklusif.
2. Berkolaborasi untuk melaksanakan tugas dengan tema komputasi.
3. Mengenali dan mendefinisikan persoalan yang pemecahannya dapat didukung dengan sistem komputasi.
4. Mengembangkan dan menggunakan abstraksi.
5. Mengembangkan artefak komputasi, misalnya membuat program sederhana untuk menunjang model komputasi yang dibutuhkan di pelajaran lain.
6. Mengembangkan rencana pengujian, menguji dan mendokumentasikan hasil uji artefak komputasi.
7. Mengkomunikasikan suatu proses, fenomena, solusi TIK dengan mempresentasikan, memvisualisasikan serta memperhatikan hak kekayaan intelektual.

Bekerja dalam kelompok untuk mengerjakan proyek Informatika mencakup hampir semua aspek tersebut, dan hasil kerja kelompok akan dipaparkan baik secara lisan, menjadi bahan presentasi, atau poster atau karya lainnya. Di zaman digital, komunikasi grafis visual memudahkan informasi tersampaikan. Oleh karena itu, kalian perlu berlatih berkomunikasi lisan, tertulis, maupun menyiapkan bahan paparan dalam bentuk grafis. Mengkomunikasikan ide dalam bahasa lisan ataupun tertulis tentunya kalian pelajari mendalam dalam pelajaran bahasa. Ini merupakan salah satu contoh, bahwa mata pelajaran bahasa sangat penting untuk menunjang informatika. Kalian harus mempraktikkan apa yang dipelajari dalam pelajaran bahasa untuk mengkomunikasikan hasil kegiatan dalam informatika.

Pada bagian ini tidak akan dibahas tentang perkakas untuk berkolaborasi, membuat bahan presentasi ataupun infografis. Bekerja dalam kelompok akan disimulasikan dalam permainan peran, presentasi akan diberikan praktik baiknya, sedangkan infografis akan diberikan arahan tentang apa yang disebut dengan infografis yang baik.

1. Kolaborasi

Kolaborasi adalah proses dua orang atau lebih yang bekerja sama untuk menyelesaikan tugas atau mencapai tujuan. Kolaborasi membutuhkan kepemimpinan, pembagian tugas dan peran yang adil. Kolaborasi yang baik akan terjadi jika komunikasi antar anggota kelompok baik.

Pada aktivitas ini, tugas diberikan untuk dikerjakan dengan kolaborasi dalam tim agar kalian dapat berdiskusi dan mengerjakan suatu tugas yang lebih besar, dibanding hanya mengerjakan seorang diri. Jumlah anggota kelompok akan ditentukan oleh guru, sesuai dengan tugas yang diberikan. Makin besar tugasnya, biasanya makin banyak anggota kelompoknya. Secara umum, tujuan kerja kelompok adalah agar kalian menghasilkan alternatif solusi dari berbagai sudut pandang, menentukan solusi yang diimplementasi secara demokratis, menyelesaikan tugas dengan lebih cepat, dan memupuk semangat bergotong royong.

Benarkah bahwa jika tugas dikerjakan oleh lebih banyak orang akan selalu lebih cepat? Belum tentu. Misalnya tugas untuk membungkus dan memasukkan 20 komputer ke dalam kardus, jika dikerjakan oleh 5 orang akan jauh lebih cepat dibandingkan dengan dikerjakan oleh 2 orang, atau bahkan 1 orang. Hal ini terjadi jika masing-masing dapat membungkus dan memasukkan ke setiap kardus sendiri-sendiri. Ada pekerjaan yang tidak semudah itu untuk dibagi. Misalnya jika hanya tersedia 2 gulungan selotip penutup kardus, dan hanya ada 1 gunting setiap orang tidak dapat bekerja mandiri karena saat membutuhkan selotip harus menunggu jika sedang dipakai. Ada juga tugas di mana setiap orang tidak mengerjakan tugas yang sama, dan harus dikerjakan dalam beberapa tahap.

a. Pemanasan: Berbagi tugas dan peran

Ayo Lakukan

Aktivitas Kelompok

Aktivitas GS-K10-01-U: Berbagi Tugas dan Peran dalam Kelompok

Aktivitas pemanasan ini dilakukan untuk memulai belajar bekerja dalam kelompok,

Deskripsi Tugas

Pada akhir kepengurusan OSIS tahun 2020, setelah pemilihan pengurus OSIS dan MPK yang baru, pengurus OSIS lama mendapat tugas menyiapkan upacara pelantikan ketua yang baru, dan setelah pelantikan akan dilakukan acara ramah tamah.



Gambar 1.5 Pelantikan OSIS (a) Upacara, (b) Nasi Tumpeng untuk acara ramah tamah

Untuk pekerjaan ini, siswa perlu berbagi peran karena setiap orang dapat mengerjakan yang tidak sama. Misalnya, karena ada dua acara, secara tegas, dapat dibentuk dua subtim yang mempunyai koordinator, dan keduanya dikoordinasikan oleh Ketua Panitia. Ada yang menyiapkan upacara pelantikan dan ada yang menyiapkan acara ramah tamah.

Ada pekerjaan yang jika dikerjakan lebih banyak orang akan membutuhkan waktu lebih lama. Oleh karena itu, strategi membagi pekerjaan dalam kelompok harus ditentukan oleh kelompok sebelum dikerjakan agar tugas dapat dikerjakan dengan lancar. Tugas juga dibagi rata secara adil walaupun setiap orang tidak mengerjakan hal yang sama. Sebelum mulai bekerja dalam kelompok, semua anggota kelompok perlu menentukan strategi yang tepat agar tujuan tugas dapat dilakukan dengan paling efisien dan optimal. Untuk tugas tertentu, perlu ditunjuk Koordinator Tim yang akan mengatur kelancaran pekerjaan, dan yang akan memeriksa apakah hasil akhir sudah sesuai dengan yang diharapkan. Untuk setiap acara, perlu disusun program kerja dan acara yang runtut berikut waktunya.

Agar bekerja dalam kelompok menjadi efisien, efektif, dan menghasilkan solusi yang baik, lakukanlah hal berikut.

1. Pahami persoalan dan tujuan bersama yang akan dicapai kelompok.
2. Lakukan pembagian peran dalam kelompok sehingga setiap anggota mendapat *peran* dan *pembagian tugas*. Jika semua mengerjakan hal yang sama, tugas tidak akan selesai. Harus ditunjuk ketua kelompok jika anggotanya banyak atau hasilnya perlu dikumpulkan dan diintegrasikan.
3. Kerjakan tugas sesuai peran masing-masing. Jangan kuatir kalau perannya berbeda. Siswa dapat berganti peran pada tugas berikutnya.
4. Setelah masing-masing mengerjakan bagian sesuai perannya, kedua subtim harus berkumpul untuk mengintegrasikan hasil kerja masing-masing.
5. Setelah subtim beres, ada koordinasi yang hanya diwakili oleh ketua subtim.
6. Kelompok dipimpin oleh ketua kelompok atau diskusi antarrekan sebaya, merumuskan kesimpulan semua kelompok dengan berdiskusi.

Apa yang kalian lakukan?

Buatlah pembagian tugas dan peran untuk setiap anggota tim. Isi Lembar Kerja Siswa ini:

Anggota	Tugas	Peran

b. Perencanaan Kerja Kelompok

Perencanaan kerja sebelum pekerjaan dilakukan akan bermanfaat bagi tim untuk mendapatkan hasil yang baik, efektif, dan efisien. Perencanaan kerja harus dilakukan dengan membuat strategi yang sesuai dengan mempertimbangkan sumber daya yang dimiliki seperti jumlah anggota tim, waktu pengerjaan, dan peralatan yang dimiliki. Walaupun pembagian perencanaan kerja ini memakan waktu, namun hal ini akan bermanfaat saat melakukan pengecekan agar pekerjaan dapat selesai tepat waktu sesuai target.

> **Tahukah kalian?**
>
> **Tugas** adalah hal yang wajib dikerjakan atau yang ditentukan untuk dilakukan; pekerjaan yang menjadi tanggung jawab seseorang; pekerjaan yang dibebankan. **Peran** adalah perilaku yang diharapkan dari seorang individu yang menempati posisi atau status sosial tertentu, contoh: ketua kelompok, bendahara, sekretaris, dan lainnya. (sumber: Kamus Besar Bahasa Indonesia (http://kbbi.kemdikbud.go.id), Ensiklopedia Britannica)

Ayo Lakukan

Merencanakan Kerja kelompok

Aktivitas GS-K10-02-U: Perencanaan Kegiatan

Latihan ini dilakukan sebelum melakukan aktivitas kelompok apapun.

Deskripsi Tugas

Kalian akan mendapat 2 atau 3 tugas yang harus dikerjakan berkelompok. Setiap kelompok harus menentukan strategi dalam mengerjakan tugas. Tugas kalian ialah membuat strategi yang paling efisien untuk menyelesaikan semua tugas yang diberikan daftarnya.

Contoh rangkaian tugas selama 1 semester dalam sebuah kelompok yang terdiri atas 6 orang, yaitu Ali, Badu, Cici, Deni, Elon, dan Fadlan:

1. Membuat maket sebuah komputer dari kardus yang dikerjakan oleh 4 orang.
2. Membuat maket sebuah ponsel dari kardus yang dikerjakan oleh 2 orang.
3. Membuat infografis tentang keamanan internet yang dikerjakan oleh 3 orang.

4. Menulis poster tentang peran teknologi dalam kehidupan sehari-hari yang dikerjakan 3 orang.
5. Membuat laporan keuangan kegiatan siswa dikerjakan oleh 3 orang.

Kalian merupakan salah satu anggota dalam kelompok tersebut dan boleh menentukan menjadi anggota kelompok yang mana. Usahakan, setiap orang mendapat beban tugas yang sama dan berkeadilan untuk ke-5 tugas secara keseluruhan.

Apa yang harus kalian lakukan?

Untuk setiap tugas, tentukan tujuan tugas, pembagian peran, langkah penyelesaian, dan tugas setiap anggota kelompok serta perkiraan waktu kerjanya [dengan mengisi *instrument* terlampir]

Proyek : ....
Anggota Kelompok : ....
Semester : ....
Proyek : ....
Anggota Kelompok : ....
Semester : ....

Aspek	Uraian
Deskripsi Umum Proyek	Tuliskan dengan ringkas judul dan tentang proyek yang kalian kerjakan.
Tujuan Tugas	Tuliskan dengan ringkas.
Hasil Tugas	Jelaskan hasil (*deliverable*) tugas dengan spesifikasi yang jelas dan terukur.
Waktu dan Tempat Penyelesaian	Berapa lama mengerjakan dan di mana (di rumah, sekolah).
Pembagian Peran	Tuliskan peran untuk setiap anggota kelompok, perannya apa.
Tahapan/Langkah Penyelesaian	Langkah umum dan runtut penyelesaiannya.
Kesimpulan Capaian	Tercapai semuanya/sebagian/ sebagian kecil saja/tidak tercapai.
Apa yang dinikmati dan disenangi dalam mengerjakan proyek ini?	Tuliskan secara ringkas perasaan kalian dan alasannya.
Kesulitan dalam mengerjakan proyek ini	Tuliskan secara ringkas kesulitan kalian dan mengapa kesulitan itu muncul. Andaikata diberi kesempatan ulang, dapatkah mengatasinya?

Kemudian setiap anggota kelompok harus membuat laporan ringkas, yang berisi hal-hal sebagai berikut, yang dituliskan bebas pada sebuah kertas dan dimasukkan dalam Buku Kerja Siswa:

- Perannya
- Apa yang sudah dikerjakan
- Kontribusinya terhadap hasil
- Refleksinya

2. Komunikasi

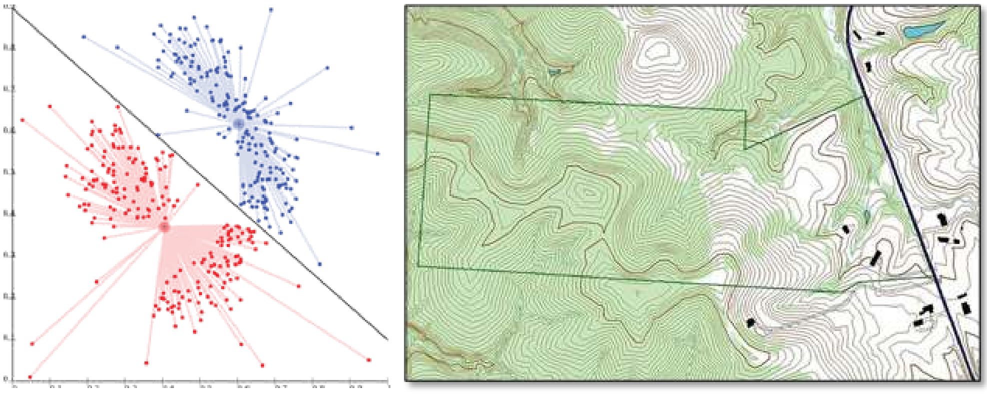
Hasil kerja yang telah kita kembangkan atau dibuat perlu dikomunikasikan dengan baik kepada pihak yang berkepentingan, seperti: pemberi kerja, khalayak umum, dan lainnya. Komunikasi hasil kerja bisa dilakukan dengan presentasi dan demo produk, atau dalam bentuk produk tampilan seperti infografis.

a. Infografis

Infografis berisi informasi teks dan gambar atau grafik lain yang akan mempermudah pembaca menangkap pesan yang dimaksud. Ilustrasi visual lebih mudah untuk ditangkap maknanya dan diinterpretasi ketimbang hanya tekstual.

Sering kali, infografis dalam 1 lembar akan lebih bermakna dan menunjukkan hal penting (disebut "abstraksi") dari apa yang kalian kerjakan. Dapat dikatakan bahwa sebuah gambar mengandung seribu makna (*a picture is worth a thousand words*)

Misalnya, Gambar 1.6 menunjukkan grafik dan infografis yang mempermudah memperoleh informasi.



(a) (b)



(c)

Gambar 1.6 Contoh grafik (a) Cluster (kelompok) Data, (b) Layer pada aplikasi Geographical Information System, (c) Infografis menjadi pengguna internet anti-hoax.



Gambar 1.7 Komik lucu tentang teknologi

Sebuah infografis mempunyai komponen konten dan tampilan. Konten dan tampilan akan menjadikan sebuah infografis menjadi paparan untuk mengomunikasikan informasi lebih baik dan berhasil.

Perancangan konten infografis perlu memperhatikan hal-hal berikut.

1. Tujuan infografis dibuat: untuk mengedukasi, himbauan, ajakan atau lainnya.
2. Kepada siapa infografis ditujukan. Misalnya, membuat infografis untuk adik-adik TK akan berbeda dengan infografis paparan proyek sains kalian. Infografis saintifik kontennya harus lebih saintifik dan sesuai kaidah sains.

3. Konten harus benar, masuk akal, dan jelas sumbernya. Konten bergantung pada tujuan infografis dibuat.

Dari segi tampilan, kalian perlu memperhatikan hal-hal berikut saat membuat infografis.

1. Pemilihan kata-kata kunci. Karena sempitnya ruangan, pemilihan kata kunci dan kalimat yang singkat dan bermakna harus diperhatikan.
2. Pemilihan huruf yang sesuai ukuran dan jenisnya.
3. Pemilihan ilustrasi (grafik, gambar) yang sesuai.
4. Tata warna yang serasi dan sesuai dengan konteks. Beberapa warna mempunyai arti tersendiri. Bahkan, kalian perlu memperhatikan komposisi warna agar teman yang buta warna akan masih bisa menginterpretasi dengan baik.
5. Tata letak yang baik.

b. Presentasi

Laporan tertulis atau sebuah artefak komputasional, yaitu produk yang akan kalian hasilkan dalam aktivitas praktik informatika, perlu dijelaskan secara lisan agar dapat lebih mudah dipahami. Untuk itu, sebelum presentasi, kalian harus menyiapkan garis besar ("outline") informasi lisan yang akan disampaikan dalam bentuk tertulis.

Sebuah bahan presentasi yang baik ialah gabungan antara penjelasan murni tekstual dan gambar atau grafik (jika dipandang per halaman presentasinya). Biasanya, bahan presentasi disampaikan dalam ringkasan butir-butir penting dari paparan (oleh sebab itu, perkakas pembuatnya disebut "PowerPoint"). Kriteria dari sebuah bahan presentasi yang baik ditentukan oleh konten dan tampilannya.

Dari segi konten:

1. Mengandung bagian penting sebagai berikut yang menunjukkan alur logika dari paparan, yang mengalir dengan runtut
   a. Ringkasan, agar dalam waktu singkat pendengar
   b. Latar belakang
   c. Paparan utama
   d. Penutup, kesimpulan
2. Ditulis dalam bahasa yang sesuai, dan dalam bentuk butir-butir kata kunci serta ringkasan teks. Seringkali bahkan bukan dalam kalimat lengkap

Dari segi tampilan:

1. Ukuran huruf cukup besar untuk dapat dilihat oleh pendengarnya.
2. Seimbang antara teks dan gambar/ilustrasi. Terlalu banyak ilustrasi mengharuskan banyak narasi. Terlalu banyak teks, akan sulit ditangkap dalam waktu singkat.

3. Jumlah lembar yang sesuai dengan waktu yang disampaikan. Makin singkat, perlu sedikit lembar presentasi dan harus padat.

Dengan bahan presentasi yang baik, kalian akan lebih percaya diri dalam melakukan presentasi atau paparan lisan. Sebaiknya, kalian memperhatikan hal-hal sebagai berikut dalam mempersiapkan bahan presentasi yang baik.

1. Siapkan bahan presentasi dengan baik dan menarik sesuai waktu yang disediakan, disertai catatan tentang penekanan apa yang akan disampaikan.
2. Siapkan alat bantu seperti alat petunjuk.
3. Perhatikan siapa pendengarnya dan konteks dari presentasi karena akan menentukan gaya bahasa lisan dan gaya bahasa tubuh yang cocok.
    a. Paparkan dengan bahasa lisan yang baik, ucapkan dengan jelas sesuai dengan bahasa yang dipakai, dan tidak menggumam.
    b. Pakai bahasa tubuh yang baik, tidak terlalu banyak gerakan mengganggu.
4. Saat menjelaskan produk, sampaikan dengan alur logika yang menunjukkan penalaran yang runtut, misalnya mulai dari deskripsi persoalan, usulan solusi, solusinya, dan kesimpulan apakah solusi mengatasi persoalan.
5. Saat tanya jawab:
    a. Jawab dengan ringkas dan tegas sesuai pernyataan.
    b. Jika tidak memahami pertanyaan, rangkum dan ulangi dulu untuk klarifikasi sebelum menjawab agar jawaban sesuai pertanyaan.

Tahukah kalian?

Bahasa tubuh adalah gerakan yang akan membantu memperjelas apa yang dipaparkan. Bahasa tubuh dilakukan dengan gerakan bagian tubuh misalnya kerdipan mata, gerakan tangan, gerakan kepala, dsb.

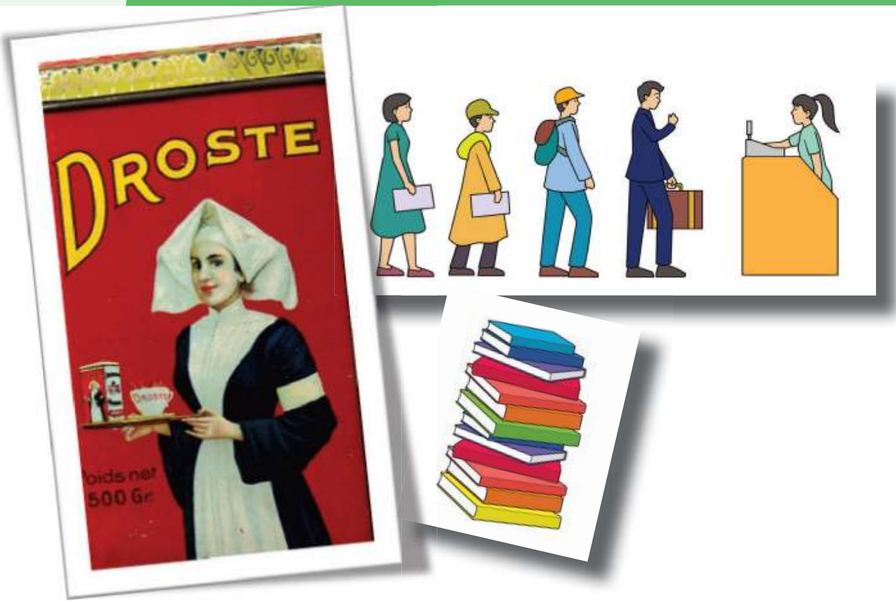
Ayo Kita Renungkan

1. Apakah kalian memahami perbedaan antara berbagi tugas dan berbagi peran? Jika belum, diskusikan dengan guru kalian.
2. Apakah setiap orang pernah mendapat peran yang sama untuk tugas yang berbeda?
3. Pelajaran apa yang kalian petik dari perencanaan setiap tugas dan perencanaan keseluruhan untuk 5 tugas?
4. Apakah kalian siap bekerja kelompok sepanjang semester?
5. Apakah menurut kalian, lebih enak jika anggota kelompok ditentukan oleh guru?

Ingat untuk membuat perencanaan sebelum mengerjakan tugas kelompok! Tuliskanlah pengalaman setiap kerja kelompok dalam jurnal kalian sehingga saat berbagi tugas, kalian bisa bergiliran dan adil.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Mushthofa, Auzi Asfarian, Dean Apriana Ramadhan,
Irya Wisnubhadra
ISBN : 978-602-244-506-7

Bab 2
Berpikir Komputasional



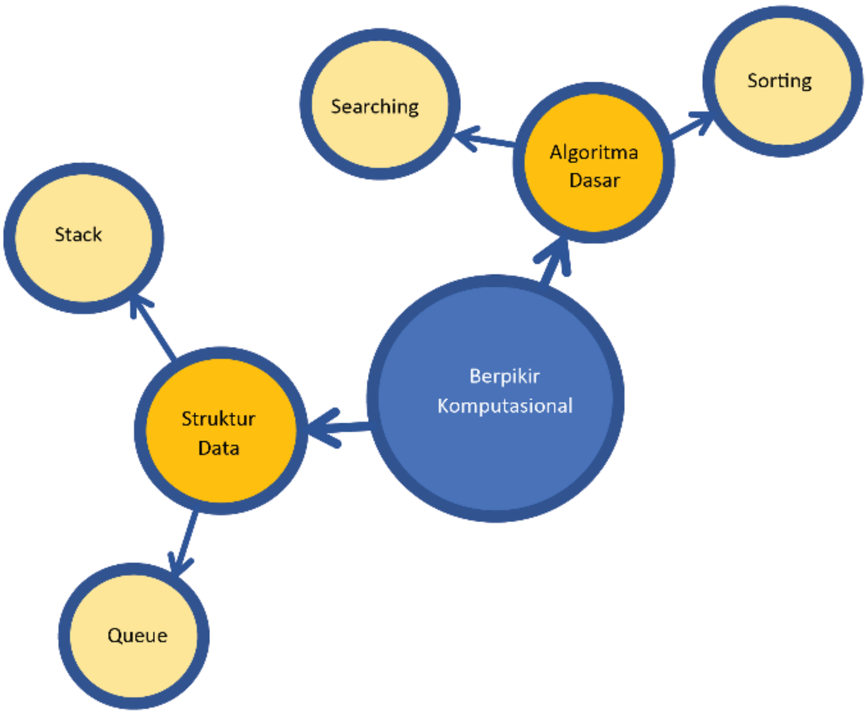
Tujuan Pembelajaran

Pada bab Berpikir Komputasional, kalian mampu menjelaskan dan menerapkan algoritma standar untuk beberapa persoalan yang disajikan, menjelaskan bagaimana data disimpan dalam struktur data tertentu, dan menentukan strategi yang efektif untuk menyelesaikan persoalan yang sajikan.

Pertanyaan Pemantik

Ketika kalian menghadapi suatu permasalahan yang harus kalian selesaikan, tentunya kalian menentukan strategi, bukan? Apa itu strategi? Mengapa dalam menemukan solusi, kalian perlu menentukan strategi?

Peta Konsep



Gambar 2.1 Peta Konsep Berpikir Komputasional

Apersepsi

Banyak peralatan dan aplikasi di dunia digital sekarang ini menggunakan komputer. Komputer dan aplikasinya dibuat oleh ahli Informatika sehingga membuat komputer tampak seperti mesin yang cerdas. Namun, benarkah komputer itu cerdas? Bagaimana komputer bisa melakukan tugas seperti manusia?

Kata Kunci

Algoritma, strategi algoritmik, *searching, sorting, stack, queue*.

Apa itu Berpikir Komputasional?

Melalui Berpikir komputasional (BK), kalian akan berlatih berpikir seperti seorang ilmuwan Informatika, bukan berpikir seperti komputer karena komputer adalah mesin.

Kegiatan utama dalam BK ialah penyelesaian masalah (*problem solving*), untuk menemukan solusi yang efisien, efektif, dan optimal sehingga solusinya bisa dijalankan oleh manusia maupun mesin. Dengan kata lain, kegiatan dalam BK ialah mencari strategi untuk mengatasi persoalan. Persoalan apa yang akan diselesaikan? Sebetulnya, hampir semua persoalan sehari-hari mengandung konsep komputasi sehingga bisa diselesaikan dengan bantuan mesin komputer. Sebagai contoh, robot yang bertugas melayani penjualan di restoran atau mengantar makanan dan obat untuk pasien di rumah sakit yang sudah dipakai di beberapa negara maju, sistem komputer untuk memantau perkebunan sawit yang siap panen dan sebagainya. Sistem komputer pada pada hakikatnya meniru dunia ini untuk dijadikan dunia digital sehingga bisa membantu atau menggantikan manusia dalam melakukan pekerjaan-pekerjaan yang sulit maupun membosankan.

Ada 4 fondasi berpikir komputasional yang dikenal dalam ilmu Informatika, yaitu Abstraksi, Algoritma, Dekomposisi, dan Pola, yang sangat mendasar dan secara garis besar dijelaskan sebagai berikut.

1. *Abstraksi*, yaitu menyarikan bagian penting dari suatu permasalahan dan mengabaikan yang tidak penting sehingga memudahkan fokus kepada solusi.
2. *Algoritma*, yaitu menuliskan otomasi solusi melalui berpikir algoritmik (langkah-langkah yang terurut) untuk mencapai suatu tujuan (solusi). Jika langkah yang runtut ini diberikan ke komputer dalam bahasa yang dipahami oleh komputer, kalian akan dapat "memerintah" komputer mengerjakan langkah tersebut.
3. *Dekomposisi* dan formulasi persoalan sedemikian rupa sehingga dapat diselesaikan dengan cepat dan efisien serta optimal dengan menggunakan komputer sebagai alat bantu. Persoalan yang sulit apalagi besar akan menjadi mudah jika diselesaikan sebagian-sebagian secara sistematis.
4. Pengenalan *pola* persoalan, generalisasi serta mentransfer proses penyelesaian persoalan ke persoalan lain yang sejenis.

BK perlu diasah dengan latihan rutin, mulai dari persoalan sederhana dan kecil. Kemudian, secara bertahap, persoalannya ditingkatkan menjadi makin besar, kompleks, dan rumit. Makin besar dan kompleks suatu persoalan, solusinya makin membutuhkan komputer agar dapat diselesaikan secara

efisien. Pada tingkat SD dan SMP, strategi penyelesaian persoalan belum secara khusus dirumuskan dalam bentuk algoritma. Pada tingkat SMA, kalian akan belajar bagaimana caranya agar solusi masalahnya bisa dituliskan dalam bentuk algoritma yang efisien dan siap dibuat menjadi program komputer.

Ruang permasalahan di dunia ini luas sekali, dan tentunya tak seorang pun ingin hidupnya menghadapi persoalan. Setiap bidang juga mempunyai persoalan dari sudut pandang bidang masing-masing, dan akan mengusulkan penyelesaian dengan menggunakan konsep dan prinsip keilmuan bidangnya. Kita belajar dari persoalan-persoalan yang ada dan pernah diusulkan solusinya. Oleh karena itu, belajar penyelesaian persoalan ialah belajar dari kasus-kasus dan solusinya. Namun, persoalan yang dibahas itu perlu di adaptasi dengan konteks kita. Kita perlu membentuk pola persoalan dan pola solusi dari latihan penyelesaiannya. Karena sangat banyak, latihan persoalan perlu dipilih. Topik yang dipilih dalam BK untuk SMA dalam mata pelajaran Informatika merupakan persoalan-persoalan mendasar terkait kehidupan sehari-hari yang perlu dikuasai dan mengandung konsep Informatika yang dominan. Bisa saja persoalan tersebut berkaitan dengan bidang lain, tetapi kita akan fokus ke aspek informatika. Memusatkan penyelesaian persoalan dari satu sudut pandang ini merupakan berpikir kritis! Dengan mempelajari dan membahas latihan-latihan pada unit pembelajaran ini, diharapkan kalian akan mendapatkan dasar pengetahuan yang diperlukan untuk menemukan solusi-solusi yang membutuhkan program komputer. Melalui kasus yang dibahas, kalian diharapkan dapat membentuk katalog solusi, yang saat dibutuhkan, akan tinggal dipakai. Melalui kegiatan BK ini, kalian menabung potongan solusi yang kelak dapat dirangkai menjadi pola solusi yang dibutuhkan untuk persoalan nyata yang dihadapi.

A. Pencarian (*Searching*)

Hidup adalah pencarian yang tiada henti. Mari, kita berpikir ke pengalaman "mencari" dalam kehidupan sehari-hari. Perhatikan contoh berikut.



1. Pernahkah kalian merasa kebingungan saat mencari sebuah buku di lemari buku kalian? Atau bahkan di perpustakaan? Saat kalian meminta bantuan kepada petugas perpustakaan, mengapa dia dapat menemukan buku yang kalian cari dengan waktu yang lebih singkat?

2. Suatu hari, kalian kehilangan baju seragam yang harus dipakai pada hari itu dan kalian mencarinya. Apa strategi kalian supaya baju tersebut cepat ditemukan?
3. Kalian mengingat sebuah potongan lirik lagu, tetapi tidak ingat judul lagu tersebut. Bagaimana kalian bisa menemukan lagu tersebut dengan cepat?

Apa itu *mencari*? Mencari adalah menemukan "sesuatu" yang bisa berupa benda, angka, konsep, informasi yang memenuhi kriteria tertentu dalam suatu ruang pencarian. Masalah pencarian sangat umum ditemukan di dalam kehidupan, termasuk dalam dunia komputasi. Ketika melakukan suatu pencarian, kalian harus menemukan suatu benda atau objek yang memenuhi kriteria tertentu dari sekumpulan benda atau objek lain. Beberapa contoh dari masalah pencarian yang sering kalian temui ialah sebagai berikut.

1. Mencari buku dengan judul tertentu di rak buku perpustakaan.
2. Mencari pakaian batik seragam kalian di lemari yang berisi semua pakaian yang kalian miliki.
3. Mencari dokumen atau web tertentu dengan mesin pencari seperti Google.

Mencari benda nyata gampang, tinggal kita lihat dan kita cocokkan dengan mata. Namun, mencari informasi atau konsep yang tidak kelihatan? Hmmmmm... Tidak mudah!



(a) (b)

Gambar 2.2. Pencarian (a) buku di perpustakaan, (b) informasi di internet

Masalah pencarian dapat dibuat dalam bentuk yang lebih formal agar dapat diterapkan pada banyak kasus. Elemen pada masalah pencarian meliputi hal-hal berikut.

1. Sekumpulan benda atau objek.
2. Kriteria dari benda atau objek yang dicari.
3. Pengecekan benda atau objek, untuk memeriksa apakah ia memenuhi kriteria pencarian.

Pertanyaan selanjutnya ialah bagaimana strategi untuk mencari. Banyak cara yang dapat kita lakukan, misalnya: kita dapat mengambil pakaian secara acak dan mengecek apakah pakaian tersebut ialah seragam batik. Cara lain, misalnya dengan memeriksa pakaian dari yang berada paling atas ke paling bawah. Tentunya, ada banyak strategi lain yang dapat kalian gunakan. Ada strategi yang lebih baik daripada strategi yang lain, bergantung pada keadaan benda atau objek tersebut saat pencarian dilakukan. Tentunya, kita akan lebih mudah mencari suatu buku dengan judul tertentu di lemari perpustakaan yang tersusun rapi dengan aturan tertentu dibandingkan dengan mencarinya di sebuah lemari yang berantakan.

Ayo Kerjakan

Aktivitas Berpasangan

Aktivitas Bk-K10-01-U: Tebak Angka

Untuk memahami masalah pencarian, kalian akan bermain tebak angka. Pada saat bermain, cobalah untuk memahami permainan tersebut dan identifikasi aspek-aspek masalah pencarian pada permainan tersebut. Carilah strategi terbaik untuk menemukan angka yang dimiliki oleh teman kalian dengan jumlah pengecekan sesedikit mungkin.

Skenario Permainan

Pada permainan ini, kalian harus berpasangan dengan salah seorang teman. Teman kalian akan memilih sebuah angka bilangan bulat antara 1 – 100 (inklusif, angka 1 dan 100 juga boleh dipilih), dan angka tersebut akan ia rahasiakan. Tugas kalian ialah menemukan angka tersebut.

Untuk menemukan angka tersebut, kalian harus mengecek apakah angka tebakan kalian ialah angka yang dimiliki oleh teman kalian. Kalian hanya bisa mengecek angka satu per satu dengan menyebutkan angka tebakan kalian tersebut.

Setiap kali kalian menebak, teman kalian harus menjawab satu dari tiga kemungkinan berikut.

1. "Benar" apabila angka yang kalian tebak sama dengan angka yang dimiliki teman kalian.
2. "Angka milikku lebih kecil" apabila angka yang dimiliki teman kalian lebih kecil dari tebakan kalian.
3. "Angka milikku lebih besar" apabila angka yang dimiliki teman kalian lebih besar dari tebakan kalian.

Tentu saja, kalian dapat menebak angka apa pun, tetapi carilah strategi yang membuat kalian dapat dengan cepat (atau dengan kata lain jumlah tebakan sesedikit mungkin) menemukan angka yang dipilih oleh teman kalian.

Catatlah angka-angka yang kalian tebak dan jumlah tebakan yang kalian lakukan di lembar kerja yang disediakan. Lakukan permainan ini minimal sebanyak dua kali. Pada permainan berikutnya, kalian bisa bertukar peran.

Ilustrasi

Berikut ini ilustrasi dari permainan di atas. Pada permainan ini, Andi dan Binti bermain secara berpasangan. Andi memilih angka, sedangkan Binti harus menebak angka tersebut.

Mula-mula, Andi memilih sebuah angka antara 1 s.d. 100 (inklusif) dan Binti mencatat tebakannya di lembar kerjanya.

Binti: "50?"
Andi: "Angka milikku lebih kecil."
Binti: "30?"
Andi: "Angka milikku lebih besar."
Binti: "40?"
Andi: "Angka milikku lebih kecil."
Binti: "35?"
Andi: "Benar!"

Pada percakapan di atas, terlihat bahwa Binti dapat menebak angka yang dipilih Andi dalam empat kali penebakkan. Percakapan tersebut dicatat dalam lembar kerja, pada permainan ke-0.

Kalian dapat memanfaatkan tabel berikut yang digunakan untuk mencatat angka yang ditebak dan berapa kali penebakan dilakukan. Pada permainan ke-0, diberikan contoh isian dari ilustrasi yang diberikan di atas. Terlihat bahwa Andi memilih angka 49. Binti memerlukan sebanyak 4 kali penebakan untuk menebak dengan benar angka yang dipilih oleh Andi, yaitu menebak secara berurutan 50, 40, 35, dan 35.

Jawabannya tentu bergantung pada tebakan berikutnya.
Siapa Andi? ....
Siapa Binti? ....

Permainan ke-	Penebakan ke-																	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	50	40	30	35														
2																		
3																		

Apa yang kalian diskusikan?

Setelah bermain, saatnya kalian memikirkan makna permainan tersebut dan cara kalian bermain. Beberapa poin diskusi yang akan kalian lakukan seperti berikut.

1. Apakah permainan ini merupakan masalah pencarian?
2. Apabila Binti menjalankan strategi yang tepat, berapa kali jumlah maksimal tebakan yang benar-benar ia perlukan?
3. Strategi pencarian seperti apa yang kalian lakukan untuk menebak sesedikit mungkin?
4. Apakah strategi kalian berbeda dengan strategi yang dilakukan teman kalian? Jika berbeda, apa perbedaannya?
5. Strategi paling bagus apa yang dapat kalian temukan untuk menemukan angka dengan jumlah tebakan paling sedikit?
6. Adakah cara lain untuk "mencari" angka yang ditebak?

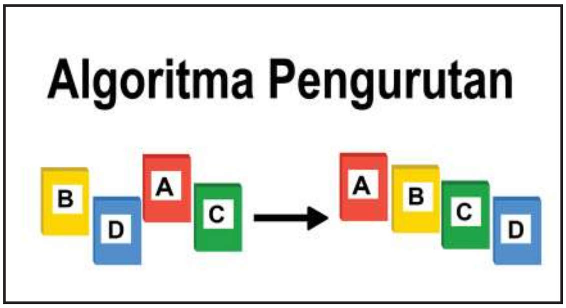
Apa yang kalian lakukan?

Tuliskan algoritma Tebak Angka dalam bahasa Indonesia. Masukkan dalam Buku Kerja Siswa.

Tantangan

Simpan tantangan ini sampai kalian sudah bisa memprogram dari modul Pemrograman. Jika kalian terpaksa melakukan permainan tebak angka sendiri karena belajar di rumah, ayo, lakukan hal ini bersama sebuah program komputer! Program komputer akan menjadi Andi, angka yang dipikirkan akan ditentukan secara acak oleh komputer. Kalian akan bisa membuat program komputer yang menjadi Andi jika kalian sudah belajar modul pemrograman. Catat hal ini sebagai salah satu hal yang akan kalian lakukan!

B. Pengurutan (*Sorting*)



Saat merapikan sesuatu, misalnya koleksi buku, kita menyusun buku tersebut dengan menggunakan suatu aturan. Misalnya, jika kita memiliki koleksi buku cerita berseri, kemungkinan besar kita akan menyusunnya secara berurut dari volume pertama hingga volume yang terbaru. Atau, ketika sedang berbaris, kita diminta untuk membentuk barisan berdasarkan tinggi badan. Hal-hal tersebut merupakan sebuah proses pengurutan atau sorting. Proses pengurutan akan

menjadi bagian yang tidak terpisahkan dari program komputer atau aplikasi yang sering kita gunakan. Pada aktivitas ini, kita akan melihat bagaimana proses pengurutan dapat dilakukan dengan menggunakan berbagai strategi. Pelajarilah strateginya!

Pengurutan merupakan suatu permasalahan klasik pada komputasi yang dilakukan untuk mengatur agar suatu kelompok benda, objek, atau entitas diletakkan mengikuti aturan tertentu. Urutan yang paling sederhana misalnya mengurutkan angka secara terurut menaik atau menurun.

Biasanya, masalah pengurutan terdiri atas sekumpulan objek yang disusun secara acak yang harus diurutkan. Setelah itu, secara sistematis, posisi objek diperbaiki dengan melakukan pertukaran posisi dua buah objek. Hal ini dilakukan secara terus-menerus hingga semua posisi objek benar.

Misal, kita memperoleh 5 buah angka acak berikut:



Kita dapat membuat angka tersebut terurut menaik dengan melakukan satu kali pertukaran, yaitu dengan menukar nilai 4 dengan nilai 3. Terdapat 2 langkah penting dalam melakukan sebuah pengurutan. Langkah pertama ialah melakukan pembandingan. Untuk melakukan pengurutan, dipastikan ada dua buah nilai yang dibandingkan. Pembandingan ini akan menghasilkan bilangan yang lebih besar dari, lebih kecil dari, atau memiliki nilai sama dengan sebuah bilangan lainnya. Langkah kedua ialah melakukan penempatan bilangan setelah melakukan pembandingan. Penempatan bilangan ini dilakukan setelah didapatkan bilangan lebih besar atau lebih kecil (bergantung pada pengurutan yang digunakan).

Terdapat beberapa teknik (algoritma) untuk melakukan pengurutan seperti *bubble sort, insertion sort, quick sort, merge sort, dan selection sort*. Pada unit ini, hanya akan diberikan penjelasan untuk setiap tiga teknik ialah sebagai berikut. Teknik lainnya dapat kalian pelajari dari referensi yang diberikan.

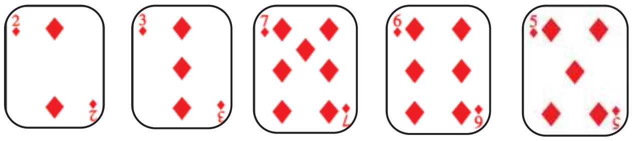
1. *Insertion Sort*

Insertion Sort adalah salah satu algoritma yang digunakan untuk permasalahan pengurutan dalam list (daftar objek). Sesuai namanya, *insertion sort* mengurutkan sebuah list dengan cara menyisipkan elemen satu per satu sesuai dengan urutan besar kecilnya elemen hingga semua elemen menjadi list yang terurut. Misalnya, dalam kasus mengurutkan elemen list dari yang terkecil hingga terbesar (*ascending*), tahap pertama ialah kita akan membaca suatu elemen dengan elemen yang berdekatan. Apabila elemen yang berdekatan dengan elemen saat ini lebih kecil, elemen yang lebih kecil akan ditukar

dengan elemen yang lebih besar dan dibandingkan kembali dengan elemen-elemen sebelumnya yang sudah terurut. Apabila elemen saat ini sudah lebih besar dari elemen sebelumnya, iterasi berhenti. Hal ini dijalankan satu per satu hingga semua list menjadi terurut.

Ilustrasi Insertion Sort

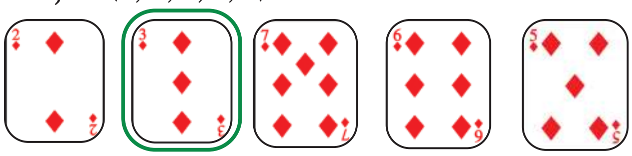
Terdapat sebuah deret bilangan seperti berikut: 2, 3, 7, 6, 5 yang direpresentasikan dengan menggunakan kartu. Urutkan bilangan tersebut secara menaik dengan menggunakan algoritma insertion sort.



Proses Iterasi Pertama

Langkah pertama, tinjau bilangan kedua, bandingkan bilangan pertama dan kedua, yaitu 2 dan 3. Didapatkan 2 lebih kecil dari 3, maka urutan bilangan tersebut tetap (2,3).

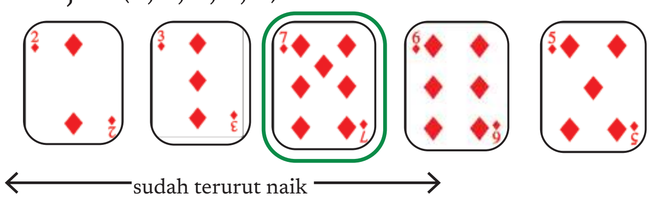
(2, **3**, 7, 6, 5) menjadi (2, **3**, 7, 6, 5)



Proses Iterasi Kedua

Pada iterasi selanjutnya, kita mengambil bilangan ketiga, yaitu 7. Lalu bandingkan dengan bilangan sebelumnya. Karena 3 lebih kecil dari 7, urutan tetap.

(2, 3, **7**, 6, 5) menjadi (2, 3, **7**, 6, 5)

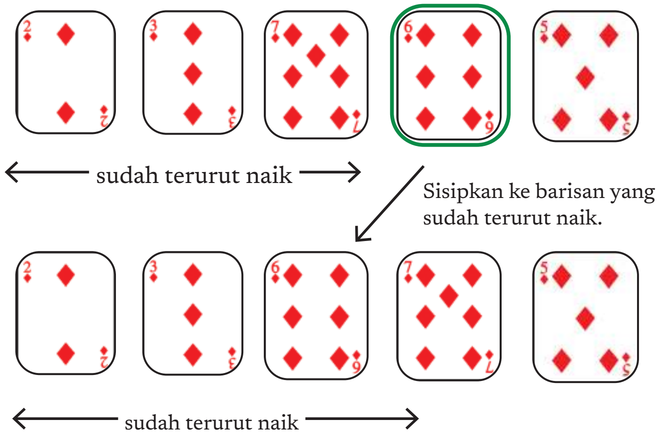


Proses Iterasi Ketiga

Pada iterasi selanjutnya, kita mengambil bilangan keempat, yaitu 6. Lalu, bandingkan dengan bilangan sebelumnya. Didapatkan bahwa 7 lebih besar dari 6. Oleh karena itu, selanjutnya, kita akan membandingkan dengan bilangan-bilangan sebelumnya, lalu menukarnya apabila bilangan tersebut lebih besar. Pertama, kita akan membandingkan 6 dan 7. Apakah 6 lebih kecil dari 7? Karena iya, kita akan menukar 6 dengan 7. Lalu, kita akan membandingkan lagi dengan bilangan sebelumnya, yaitu 3. Apakah 6 lebih kecil dari 3? Karena 6 tidak lebih kecil dari 3, maka 6 sudah berada pada posisi yang benar, yaitu sebelum 7 dan setelah 3.

Proses memindahkan 6 di antara 3 dan 7 ini biasa disebut penyisipan (insertion) sehingga nama algoritma ini disebut insertion sort.

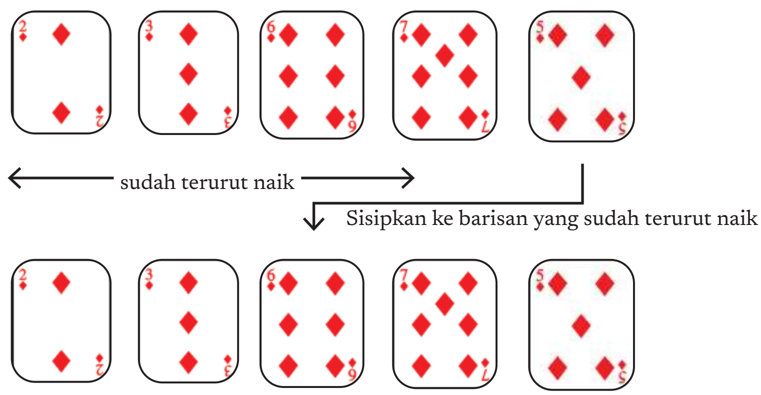
(2, 3, **7**, **6**, 5) menjadi (2, 3, **6**, **7**, 5)



Proses Iterasi Keempat

Pada iterasi selanjutnya, kita mengambil bilangan kelima, yaitu 5. Didapatkan bahwa 7 lebih besar dari 5. Oleh karena itu, selanjutnya, kita akan membandingkan dengan bilangan-bilangan sebelumnya, lalu menukarnya apabila bilangan tersebut lebih besar. Pertama, kita akan membandingkan 5 dan 6. Apakah 5 lebih kecil dari 6? Karena iya, kita akan menukar 5 dengan 6. Setelah itu, kita akan mengecek dengan bilangan sebelumnya lagi, yaitu 3. Apakah 5 lebih kecil dari 3? Karena 5 tidak lebih kecil dari 3, maka 5 sudah pada posisi seharusnya, yaitu setelah 3 dan sebelum 6. Terjadi lagi proses penyisipan kartu 5 di antara 3 dan 6.

(2, 3, 6, 7, **5**) menjadi (2, 3, **5**, 6, 7)



3. *Selection sort*

Selection sort merupakan algoritma pengurutan yang juga cukup sederhana, dengan algoritma mencari (menyeleksi) bilangan terkecil/terbesar (bergantung pada urut naik atau turun) dari daftar bilangan yang belum terurut dan

meletakkannya dalam daftar bilangan baru yang dijaga keterurutannya. Algoritma ini membagi daftar bilangan menjadi dua bagian, yaitu bagian terurut dan bagian yang belum terurut. Bagian yang terurut di sebelah kiri dan bagian yang belum terurut di sebelah kanan. Awalnya, semua elemen bilangan dalam daftar ialah bagian yang belum terurut, dan bagian yang terurut kosong. Berikut langkah-langkah yang terdapat pada algoritma *selection sort*.

1. Cari bilangan terkecil yang ada pada bagian belum terurut.
2. Tukar bilangan tersebut dengan bilangan pertama bagian belum terurut, lalu masukkan ke bagian terurut.
3. Ulangi langkah 1 dan 2 sampai bagian yang belum terurut habis.

Ilustrasi urut-urutan *selection sort* dapat dilihat pada tabel berikut.

Bagian terurut	Bagian yang belum terurut	Nilai terkecil dari bagian belum terurut
()	(2,3,7,6,5)	2
(2)	(3,7,6,5)	3
(2,3)	(7,6,5)	5
(2,3,5)	(6,7)	6
(2,3,5,6)	(7)	7
(2,3,5,6,7)	()	

Secara rinci, algoritma *selection sort* yang dikaitkan dengan pemrograman dijelaskan sebagai berikut.

Terdapat sebuah daftar bilangan tidak terurut seperti berikut: 2, 3, 7, 6, 5. Urutkan bilangan tersebut secara menaik dengan menggunakan algoritma *selection sort.*

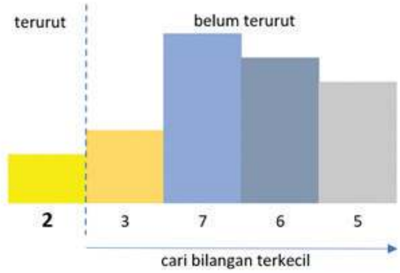
Proses Iterasi Pertama

Data Awal:



Cari bilangan terkecil di bagian belum terurut: ditemukan 2 sebagai bilangan terkecil.

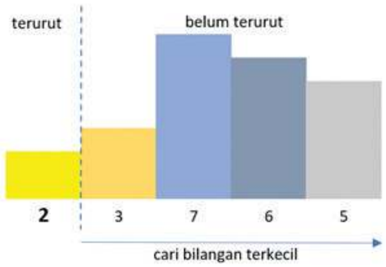


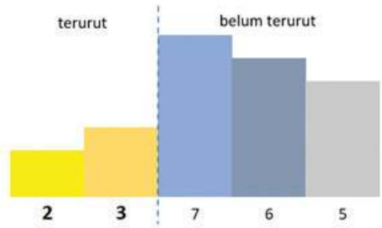


Tukar bilangan 2 dengan bilangan pertama bagian belum terurut. Geser batas bagian yang sudah terurut ke kanan sehingga 2 menjadi bagian yang sudah terurut. Dalam ilustrasi ini, angka yang dicetak tebal menunjukkan bilangan yang sudah terurut.

Proses Iterasi Kedua

Cari bilangan terkecil di bagian belum terurut, ditemukan angka 3 sebagai bilangan terkecil.

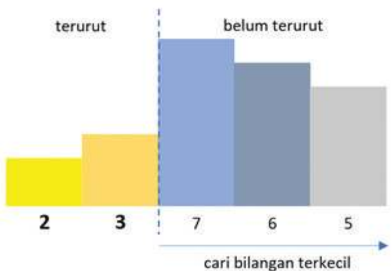


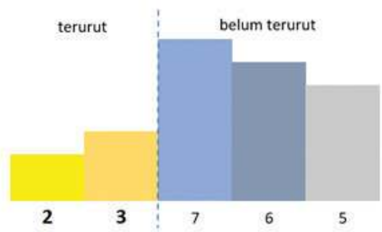


Tukar bilangan 3 dengan bilangan pertama bagian belum terurut. Geser batas bagian yang sudah terurut ke kanan sehingga 3 menjadi bagian yang sudah terurut.

Proses Iterasi Ketiga

Cari bilangan terkecil di bagian belum terurut, ditemukan angka 5 sebagai bilangan terkecil.

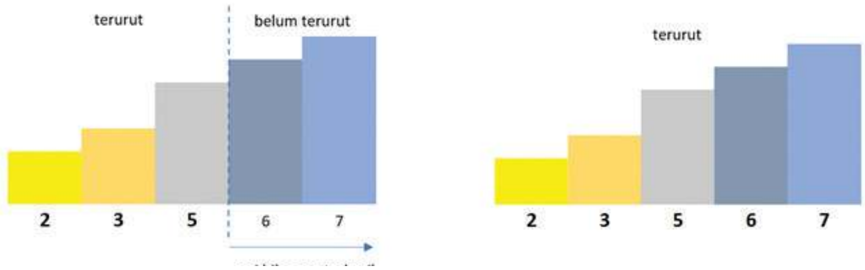




Tukar bilangan 5 dengan bilangan pertama bagian belum terurut, yaitu 7. Geser batas bagian yang sudah terurut ke kanan, sehingga 5 menjadi bagian yang sudah terurut.

Proses Iterasi Keempat

Cari bilangan terkecil di bagian belum terurut, ditemukan angka 6 sebagai bilangan terkecil.



Tukar bilangan 6 dengan bilangan pertama bagian belum terurut. Di bagian akhir, karena data tinggal dua, setelah proses penukaran, algoritma telah selesai dilaksanakan.



Aktivitas Individu/Berkelompok

Aktivitas BK-K10-02: Bermain Kartu

Apa yang kalian perlukan?

10 kartu yang masing-masing bertuliskan angka 1 sampai 10.



Skenario Permainan

Aktivitas dapat dilakukan secara mandiri atau berkelompok.

1. Kalian akan diberikan sebuah kartu bertuliskan angka dari 1 - 10.
2. Kelima belas kartu tersebut kalian kocok dan letakkan dalam bentuk barisan di atas meja. Kartu diletakkan tertutup.
3. Kalian harus dapat mengurutkan semua kartu secara menaik. Kartu yang berada di paling kiri barisan harus yang paling kecil.
4. Untuk mengurutkan, kalian harus melakukan serangkaian pertukaran kartu. Pertukaran dilakukan dengan membuka dua buah kartu. Apabila diperlukan, kalian dapat menukar posisi kedua kartu tersebut.
5. Kalian diminta untuk menyusun algoritma pertukaran yang dapat dilakukan untuk memastikan semua kartu dalam posisi terurut. Kalian dapat memilih untuk menggunakan salah satu dari tiga algoritma pengurutan yang disampaikan pada bagian konsep.

Ayo Berdiskusi

Apa yang kalian diskusikan?

Setelah bermain, saatnya memikirkan permainan tersebut dan cara kalian bermain. Beberapa poin yang penting untuk didiskusikan seperti berikut.

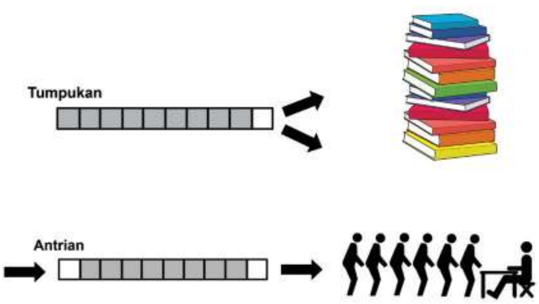
1. Apakah permainan tadi merupakan masalah pengurutan?
2. Strategi pengurutan seperti apa yang kalian lakukan untuk melakukan pengecekan dan pertukaran sesedikit mungkin?
3. Apakah strategi kalian berbeda dengan strategi yang dilakukan oleh teman kalian? Jika berbeda, apa perbedaannya?
4. Strategi paling bagus apa yang dapat kalian temukan untuk mengurutkan dengan banyaknya pertukaran paling sedikit?
5. Adakah kondisi yang membuat kalian melakukan banyak sekali pertukaran untuk mengurutkan kartu secara menaik?

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Apakah kalian sudah pernah melakukan permainan ini?
2. Saat mengurutkan kartu, apakah kalian senang?
3. Apakah kalian paham bahwa mengurutkan kartu itu suatu proses pengurutan?
4. Apakah kalian berhasil menemukan cara yang paling cepat untuk mengurutkan kartu tersebut?
5. Apakah kalian merasa ada masalah lain yang serupa dengan permainan tadi?
6. Pelajaran paling berkesan apa yang kalian dapatkan dari permainan ini?

C. Tumpukan (*Stack*) dan Antrean (*Queue*)



Kita akan mempelajari dua buah konsep cara penyimpanan data/objek dalam sebuah struktur yang akan menentukan urutan pemrosesan data/objek tersebut, yaitu tumpukan (stack) dan antrean (queue). Kedua konsep ini memiliki prosedur yang berbeda dalam menyimpan dan mengeluarkan data. Kedua konsep tersebut masing-masing memiliki peranan yang berbeda dan digunakan pada situasi yang berbeda pula.

Bayangkan sebuah loket di sebuah rumah sakit, di mana para pasien yang akan berobat diminta untuk mendaftar lebih dahulu di loket penerimaan serta mengisi formulir pendaftaran. Setelah formulir tersebut diisi, para pasien akan mengembalikan formulir ke loket dan menunggu dipanggil oleh petugas. Kebetulan, di pagi hari, dokter yang bertugas belum datang sehingga para pasien harus menunggu. Ketika sang dokter tiba, petugas loket akan memanggil para pasien satu per satu untuk mendapat layanan.

Perhatikan sekarang bagaimana urutan pasien itu dipanggil oleh petugas loket.

1. Misalkan, petugas loket menumpuk formulir-formulir tersebut di mana formulir yang baru diterima diletakkan *di atas* formulir yang sudah diterima sebelumnya, kemudian ketika ketika memanggil pasien, petugas tersebut memanggil dengan urutan mulai dari formulir yang berada *di atas* tumpukan. Menurut kalian, apakah urutan tersebut adil/sesuai dengan yang diharapkan para pasien? Mengapa?
2. Bagaimana cara petugas menyusun tumpukan formulir dan/atau cara urutan memanggil para pasien dari tumpukan formulir sedemikian rupa sehingga pasien yang datang dan mengisi formulir lebih dulu, akan *dipanggil lebih dulu* juga (dan sebaliknya)?

Dalam dunia komputasi/informatika, terkadang, kita perlu untuk menyimpan data/objek dalam suatu urutan tertentu, untuk kemudian/sewaktu-waktu diambil/dikeluarkan kembali, mungkin untuk diproses lebih lanjut atau untuk tujuan-tujuan lain. Ada dua cara utama kita dapat melakukan penyimpanan ini.

1. Antrean (*queue*): pada metode ini, objek-objek disimpan dalam metode penyimpanan yang berupa sebuah antrean sehingga objek yang pertama/lebih dulu datang, juga akan lebih dulu keluar/selesai, layaknya sebuah antrean di loket, pintu masuk, dll. Prinsip ini disebut prinsip *First In First Out* (FIFO). Dalam sebuah antrean orang, misalnya, jelas orang yang pertama datang akan berada di depan antrean, dan harus menjadi yang pertama yang mendapat pelayanan.
2. Tumpukan (*stack*): pada metode ini, objek-objek disimpan dalam metode penyimpanan yang menyerupai sebuah tumpukan (misal: tumpukan piring). Dengan demikian, objek yang *pertama/lebih dulu* disimpan justru akan menjadi yang terakhir keluar. Prinsip ini disebut juga *Last In First Out* (LIFO). Dalam tumpukan piring, misalnya, piring pertama yang diletakkan akan berada di posisi paling bawah, dan jika kita ambil piring satu per satu dari tumpukan itu, tentunya piring yang berada di posisi paling bawah tersebut akan menjadi yang terakhir diambil.

Baik dalam kehidupan sehari-hari maupun dalam dunia informatika, kedua konsep urutan penyimpanan data tersebut memiliki peran dan kegunaan masing-masing. Ada permasalahan-permasalahan/situasi di mana antrean (FIFO) lebih cocok digunakan. Sebaliknya, ada juga permasalahan-permasalahan di mana tumpukan (LIFO) lebih tepat diterapkan. Untuk lebih memahami kedua konsep ini dan bagaimana mereka digunakan, mari, kita lakukan beberapa aktivitas di bawah ini.

Ayo Kerjakan

Aktivitas Berkelompok

Aktivitas BK-K10-03-U: Penggunaan *Stack* dan *Queue* secara Tepat (*Unplugged*)

Pada aktivitas ini, kalian akan membaca beberapa skenario kondisi, baik dalam dunia sehari-hari maupun dalam dunia informatika. Tugas kalian ialah memikirkan, pada setiap kondisi/skenario tersebut, manakah yang lebih tepat digunakan/lebih relevan menggambarkan situasi tersebut, apakah stack ataukah queue. Berikan penjelasan mengapa kalian memilih jawaban tersebut!

1. Di persimpangan jalan, terdapat lampu merah. Apabila lampu merah menyala, mobil-mobil yang datang ke persimpangan tersebut harus berhenti dulu. Ketika lampu berubah menjadi hijau, semua mobil perlahan-lahan berjalan kembali dalam urutan tertentu. Manakah yang lebih tepat menggambarkan situasi tersebut?
2. Ketika menjelajah web/internet, kita menggunakan sebuah *browser* (misal Firefox, Chrome dll). Terdapat sebuah fitur yang memungkinkan kita untuk bergerak dari satu halaman yang sudah kita kunjungi ke halaman lainnya, yaitu dengan menekan tombol *Back* dan *Forward*. Misalnya, kita mengunjungi halaman A, kemudian B, lalu C. Jika kita kemudian menekan tombol *Back*, dari halaman C kita akan kembali ke halaman B. Jika kita tekan lagi tombol Back (pada saat ada di B), kita akan kembali ke A. Jika kemudian kita tekan tombol *Forward*, kita akan kembali halaman B, dan jika kita tekan sekali lagi tombol Forward, kita akan kembali ke halaman C. Oleh karena itu, aplikasi browser tersebut harus menyimpan (dan mengingat) semua halaman yang sudah pernah kita kunjungi sebelumnya (biasa disebut Riwayat atau *History*). Bentuk penyimpanan yang manakah (*stack* atau *queue*) yang paling tepat digunakan untuk menyimpan Riwayat pada *browser*?



Gambar 2.3 Tombol Back dan Forward pada Firefox
Sumber: Dokumen Kemendikbud, 2021

3. Mesin *printer* bertugas untuk mencetak dokumen yang dikirimkan dari sebuah komputer. Satu buah printer dapat terhubung ke beberapa buah komputer sekaligus, dan semuanya dapat mengirim perintah kepada *printer* tersebut untuk mencetak dokumen yang berbeda-beda. *Printer* tersebut tentunya hanya bisa mencetak satu buah dokumen dalam satu waktu tertentu, dan mungkin membutuhkan beberapa detik/menit untuk menyelesaikan proses cetak satu dokumen. Oleh karena itu, ketika *printer* sedang sibuk mencetak sebuah dokumen dari sebuah komputer, kemudian datang permintaan mencetak dari beberapa komputer yang lain (yang berbeda). *Printer* tersebut harus menyimpan dokumen-dokumen yang baru datang tersebut agar nanti dapat dicetak ketika proses pencetakan yang sedang berjalan saat ini sudah selesai. Manakah yang lebih tepat digunakan, *stack* atau *queue* untuk penyimpanan dokumen-dokumen yang sedang "menunggu giliran" untuk dicetak tadi?
4. Pada sebuah aplikasi pengolah dokumen, biasanya terdapat fasilitas untuk melakukan *Undo* dan *Redo*. Operasi *Undo* akan membatalkan langkah/ tindakan terakhir yang kita lakukan saat mengedit dokumen (misal, jika kita menyadari ada kesalahan pada langkah terakhir kita), sedangkan *Redo* digunakan untuk mengulang kembali operasi yang baru saja dibatalkan dengan sebuah *Undo*. Proses *Undo* dan *Redo* ini dapat dilakukan sampai dengan operasi pertama setelah sebuah dokumen dibuka/disimpan. Misalnya, terjadi rangkaian kejadian berikut:
   a. Budi membuka dokumen A
   b. Budi menambahkan judul pada dokumen A
   c. Budi menulis sebuah paragraf pada dokumen A
   d. Budi menambahkan sebuah tabel pada dokumen A
   e. Budi menyisipkan sebuah gambar pada dokumen A

Apabila kemudian Budi menekan tombol *Undo*, operasi terakhir (yaitu penambahan gambar) akan dibatalkan sehingga gambar tersebut akan hilang dari dokumen. Jika kemudian Budi menekan tombol *Undo* sekali lagi, operasi terakhir sebelum itu (yaitu menambahkan tabel) juga akan dibatalkan sehingga tabel tersebut akan hilang dari dokumen. Jika kemudian Budi menekan tombol *Redo*, operasi *Undo* yang terakhir (yaitu yang menghilangkan tabel) akan dibatalkan sehingga tabel tersebut akan muncul kembali.

Jelas bahwa aplikasi perlu untuk menyimpan data-data berupa tindakan/ operasi apa saja yang dilakukan oleh penggunanya dari awal sampai akhir, serta efeknya terhadap dokumen agar dapat memberikan fungsionalitas *Undo* dan *Redo* tersebut. Manakah di antara *stack* dan *queue* yang lebih tepat digunakan untuk menyimpan operasi-operasi tersebut?

Lembar Kerja Siswa

Untuk setiap kasus di atas, lakukan analisis penggunaan stack dan queue dengan mengisi LKS ini.

Persoalan	Stack	Queue	Saya Pilih ...
Persimpangan lampu merah			
Penjelajahan internet			
Antrean permintaan print dokumen dalam sebuah komputer			
Undo Redo			

Ayo Lakukan

Aktivitas Berpasangan

Aktivitas BK-K10-04-U: Simulasi Stack

Pada aktivitas ini, kalian akan bermain dengan satu orang siswa lainnya. Satu orang harus berperan menjadi *Pemberi Perintah* dan satu lagi harus berperan sebagai *Simulator*. Permainan dimulai dengan Pemberi Perintah memberikan sebuah perintah simulasi (yang akan dijelaskan di bawah). Kemudian, Simulator harus menjalankan simulasi dan memberikan jawaban yang benar. Jawaban tersebut harus diperiksa oleh Pemberi Perintah dan kemudian harus dinyatakan jawaban tersebut benar atau salah. Setelah itu, kedua orang bertukar peran: Simulator harus menjadi Pemberi Perintah dan Pemberi Perintah menjadi Simulator. Lakukan pertukaran ini sampai beberapa kali. Orang yang berhasil mendapatkan jawaban benar sebanyak mungkin akan menjadi pemenangnya.

Berikut ini format/bentuk perintah serta bentuk jawaban yang diinginkan.

Kita asumsikan ada sebuah *stack* yang mampu menyimpan nilai-nilai bilangan. Setiap perintah simulasi berisi kumpulan dari 2 buah perintah:

1. Push X
2. Pop

Perintah Push digunakan untuk menyimpan nilai ke dalam *stack*. Perintah ini harus diikuti oleh sebuah bilangan bulat X yang akan disimpan ke dalam *stack*. Perintah Pop digunakan untuk mengeluarkan angka yang *berada di atas tumpukan* saat ini. Jika saat ini tumpukan kosong, perintah Pop tidak memberikan efek apa-apa. Berikut ini contoh sebuah perintah simulasi dan hasilnya:

Perintah	Isi Stack	Hasil Pop
Push 5	5	
Push 3	5,3	

Perintah	Isi Stack	Hasil Pop
Push 2	5,3,2	
Push 4	5,3,2,4	
Pop	5,3,2	4
Push 6	5,3,2,6	
Pop	5,3,2	6
Pop	5,3	2
Pop	5	3

Ketika seorang Simulator menerima sebuah perintah simulasi, ia harus memberikan jawaban berupa daftar *bilangan yang akan dikeluarkan* dari *stack*, sesuai dengan urutan perintah simulasi yang ia terima. Misalnya, pada contoh di atas, Simulator harus memberikan jawaban berupa:

4
6
2
3

Tentunya, banyaknya angka pada jawaban harus sama dengan banyaknya perintah Pop yang diberikan oleh simulator.

Lembar Kerja Siswa

Untuk permainan peran ini dapat dipakai LKS berikut ini.

Pemberi Perintah	Catatan Simulator	Isi Stack	Hasil Pop

Jawaban Simulator:

Aktivitas BK-K10-05-U: Simulasi *Queue*

Pada aktivitas ini, kalian akan melakukan simulasi operasi pada sebuah *queue*. Serupa dengan aktivitas sebelumnya, aktivitas ini dijalankan oleh dua orang yang akan bertugas sebagai Pemberi Perintah dan Simulator. Pemberi Perintah akan memberikan kumpulan perintah yang berisi operasi pada *queue*, sedangkan Simulator harus memberikan jawaban berupa rangkaian isi *queue* yang dihasilkan dari setiap perintah yang diberikan.



Format perintah ialah sebagai berikut.

1. Enqueue X: memasukkan sebuah bilangan bulat ke dalam *queue.*
2. Dequeue: membuang/mengeluarkan bilangan yang berada pada posisi pertama antrean.

Untuk setiap perintah, Simulator harus menuliskan *apa isi queue* setiap kali perintah tersebut selesai dijalankan. Sebagai contoh, Pemberi Perintah memberikan perintah-perintah sebagai berikut.

Perintah	Simulator menulis isi queue setelah setiap perintah dijalankan	Hasil Dequeue
Enqueue 5	5	
Enqueue 3	5, 3	
Dequeue	3	5
Enqueue 4	3, 4	
Dequeue	4	3

Jika Simulator harus memberikan 5 baris jawaban berupa isi dari *queue* setelah setiap perintah dijalankan, hasilnya:

1. 5
2. 5, 3
3. 3
4. 3, 4
5. 4

Lembar Kerja Siswa

Untuk permainan peran ini dapat dipakai LKS.

Pemberi Perintah	Catatan Simulator	Isi *Stack*	Hasil Pop

Jawaban Simulator:

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Apakah kalian dapat memahami dengan baik perbedaan dari konsep *stack* dan *queue*?
2. Jika diberikan sebuah kondisi di dunia nyata/informatika, dapatkah kalian menentukan apakah stack atau queue yang lebih relevan diterapkan sebagai metode penyimpanan?

3. Dapatkah kalian mencari contoh-contoh lain penerapan stack dan queue dalam kehidupan sehari-hari?
4. Apakah kalian dapat memainkan permainan simulasi stack dan queue di atas dengan baik? Apakah permainan tersebut membantu proses pemahaman kalian terhadap kedua konsep tersebut??

Ingin Tahu Lebih?

Jika ingin belajar lebih mendalam tentang materi di atas, kalian bisa mengunjungi tautan berikut ini.

Pencarian (*Searching*)

1. Search Algorithm: https://en.wikipedia.org/wiki/Search_algorithm
2. Binary Search: https://khanacademy.org/computing/computer-science/algorithms/binary-search

Pengurutan (*Sorting*)

1. Sorting Algorithm: https://en.wikipedia.org/wiki/Sorting_algorithm
2. Video Bubble Sort:https://youtu.be/nmhjrI-aW5o
3. Video Insertion Sort: https://youtu.be/OGzPmgsI-pQ
4. Video Selection Sort:https://youtu.be/xWBP4lzkoyM

Rekursi

1. Rekursi: https://en.wikipedia.org/wiki/Recursion

Graf:

1. Graf: https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)
2. Penelusuran graf: https://en.wikipedia.org/wiki/Graph_traversal

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Wahyono, Irya Wisnubhadra
ISBN : 978-602-244-506-7

Bab 3
Teknologi Informasi dan Komunikasi



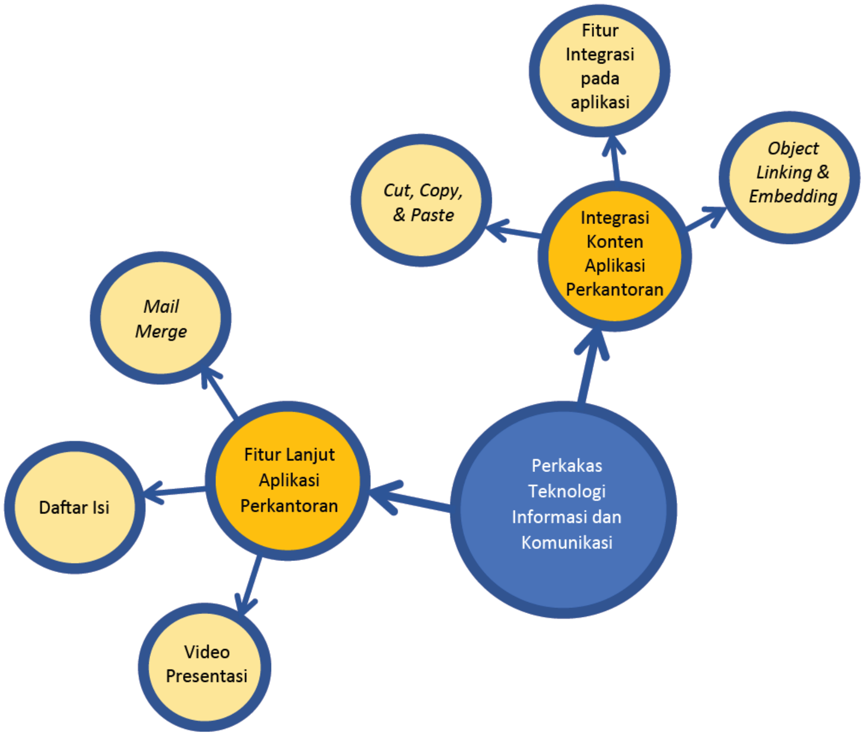
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu mengintegrasi konten antaraplikasi perkantoran (pengolah kata, angka, dan presentasi), menggunakan fitur lanjut, dan menyajikan konten aplikasi dalam berbagai representasi yang mudah untuk dianalisis.

Pertanyaan Pemantik

Tahukah kalian cara menggabungkan konten hasil beberapa aplikasi?

Peta Konsep



Gambar 3.1 Peta Konsep Teknologi Informasi dan Komunikasi

Apersepsi

Ada kalanya, kalian perlu membuat sebuah laporan yang membutuhkan data dari berbagai file aplikasi perkantoran yang kita miliki. Oleh sebab itu, integrasi antaraplikasi perkantoran ini sangat diperlukan. Hal ini untuk memudahkan kalian dalam proses analisis data sehingga penyajiannya jadi lebih baik. Bayangkan kalau kalian harus menyajikan data secara langsung dari file yang berbeda-beda. Tentunya akan sangat tidak nyaman.

Kata Kunci

Integrasi aplikasi perkantoran, fitur lanjut aplikasi perkantoran, Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Mail Merge, Object Linking & Embedding, Video Presentasi

Aplikasi office (aplikasi perkantoran) adalah kumpulan aplikasi yang digunakan untuk memudahkan pekerjaan perkantoran. Karena berupa aplikasi, fitur-fiturnya sudah tersedia dan dapat langsung digunakan oleh pengguna, tanpa harus membangunnya sendiri menggunakan bahasa pemrograman.

Secara umum, aplikasi perkantoran terbagi menjadi beberapa jenis menurut kegunaan dan fungsinya. Tiga di antaranya yang paling banyak digunakan ialah seperti berikut.

1. Aplikasi pengolah kata (*word processor*). Aplikasi ini fitur utamanya digunakan untuk membuat dan mengolah dokumen, misalnya membuat surat, menyusun laporan, dan lain-lain. Contoh aplikasi pengolah kata ialah Microsoft Word, Open Office Word, Google Docs, dan lain-lain.
2. Aplikasi pengolah lembar kerja (*spreadsheet*). Aplikasi ini fitur utamanya digunakan untuk mengelola data yang disajikan dalam bentuk lembar kerja, yaitu tabel dua dimensi yang terdiri atas kolom dan baris fitur-fitur yang tersedia dapat mempermudah perhitungan data dan mengolah data secara statistik, dan dapat menampilkannya dalam bentuk diagram. Contoh aplikasi spreadsheet ialah Microsoft Excel, OpenOffice Spreadsheet, Google Sheet, dan lain lain.
3. Aplikasi pembuat bahan presentasi yang disebut *slide*. Aplikasi ini fitur utamanya digunakan untuk memudahkan kita untuk membuat slide presentasi dengan mudah dan cepat serta menghasilkan *slide* yang menarik. *Slide* presentasi adalah objek yang terdiri atas butir-butir penjelasan, dapat disertai gambar atau ilustrasi lainnya. Contoh perangkat lunak pembuat presentasi ialah Microsoft Office PowerPoint, OpenOffice Presentation, Google Slide, dan lain-lain.

A. Integrasi Konten Aplikasi Perkantoran

Mengintegrasikan artinya menyatukan beberapa objek, data, atau komponen untuk membentuk sesuatu yang utuh dan bermakna, walaupun komponennya tadinya terpencar atau tidak ada hubungannya. Misalnya, kita harus membuat sebuah laporan berdasarkan pengamatan yang datanya dibuat dan divisualisasi dengan excel. Sebelum laporan lengkap, kita membuat proposal dalam bentuk *slides*. Saat kita membuat laporan, untuk menghindari mengerjakan hal yang sama (mengetik ulang), kita perlu "membawa" potongan hasil kerja dengan MS Excel, MS Word, dan MS PowerPoint ke dalam MS Word karena laporan akhir dibuat dengan MS Word.

Setiap jenis aplikasi perkantoran memiliki fungsi yang berbeda sehingga pengguna biasanya hanya menggunakan satu aplikasi sekali waktu, misalnya menggunakan aplikasi word processor untuk menyusun dokumen. Akan

tetapi, ada kalanya, pengguna memerlukan lebih dari satu aplikasi untuk suatu keperluan. Sebagai contoh, saat diperlukan laporan yang memuat diagram, selain aplikasi pengolah kata, diperlukan juga aplikasi pengolah lembar kerja yang dapat membuat diagram dengan lebih presisi. Oleh karena itu, diperlukan integrasi konten antaraplikasi perkantoran.

Integrasi memungkinkan untuk mengaitkan data maupun fitur antaraplikasi sehingga data atau fitur pada satu aplikasi dapat digunakan di aplikasi yang lain. Integrasi konten aplikasi perkantoran bertujuan untuk menggabungkan konten aplikasi lain dalam satu aplikasi. Sebagai contoh, pada Microsoft Office, data dari Microsoft Excel dapat ditautkan dengan dokumen pada Microsoft Word dalam pembuatan surat, atau data di Microsoft Excel ditautkan dengan presentasi pada MS PowerPoint dalam menampilkan grafik dan tabel. Kedua contoh tersebut merupakan contoh implementasi dari integrasi konten antaraplikasi di Microsoft Office. Microsoft Excel digunakan sebagai sumber data atau objek untuk disajikan dalam dokumen atau presentasi.

Terdapat beberapa cara untuk mengintegrasikan data, teks, gambar antar aplikasi perkantoran. Cara itu menggunakan: (a) teknik "Salin-Tempel"(*Copy Paste*) atau "Gunting-Tempel" (*Cut Paste*), (b) Menu yang tersedia di aplikasi, (c) teknik *Object Linking dan Embedding*.

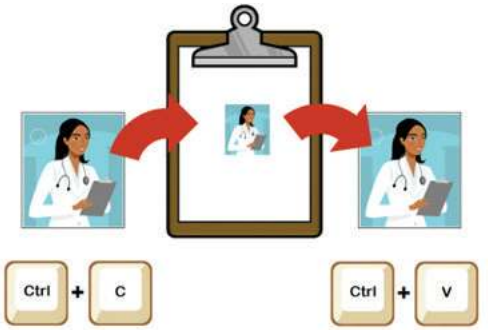
1. Integrasi dengan Perintah *Cut, Copy,* dan *Paste*

Perintah *Cut* (Potong), *Copy* (Salin), dan *Paste (*Tempel) pada aplikasi komputer saat ini diinspirasi dari praktik tradisional dalam pengeditan naskah yang diketik pada sebuah kertas, di mana orang akan memotong (*cut*) paragraf dari halaman dengan gunting dan menempelkannya ke halaman lain (*paste*). Praktik ini tetap berlangsung hingga 1980-an. Pada saat itu, toko alat tulis bahkan menjual "gunting pengeditan" dengan bilah cukup panjang yang mampu memotong halaman selebar 22 cm.

Saat ini, perintah *Cut, Copy,* dan *Paste* sangat populer digunakan. Banyak aplikasi menyediakan cara unik untuk metode ini seperti: kombinasi tombol, menu tarik-turun (*pull-down* menu), menu *pop-up,* dll.

Mekanisme tradisional *Cut, Copy,* dan *Paste* menggunakan gunting, sedangkan pada aplikasi komputer, perintah *Cut* memindahkan teks/grafik atau objek lain ke dalam *clipboard* atau *buffer* berupa tempat penyimpan sementara. Perintah *Paste* akan memindahkan objek dari *clipboard* tersebut menuju ke dokumen tujuan. Perintah *Copy* akan menyalin teks/grafik atau objek lain yang disorot ke dalam *clipboard* dan akan memindahkan objek dari *clipboard* tersebut menuju ke dokumen tujuan.

Perintah "Potong dan Tempel" memiliki urutan cara yang sama, tetapi perintah untuk Potong, yaitu menggunakan Ctrl-x (⌘ + x untuk pengguna Macintosh). Contohnya, melakukan *Copy-Paste* pada tabel dari aplikasi lembar kerja ke pengolah kata. Atau, melakukan penyalinan dari MS Word dan ditempel ke MS Excel atau MS PowerPoint. Artinya, kita dapat membuka 2 atau 3 aplikasi sekaligus, dan membawa potongan teks, tabel, gambar dari satu aplikasi ke aplikasi lainnya.

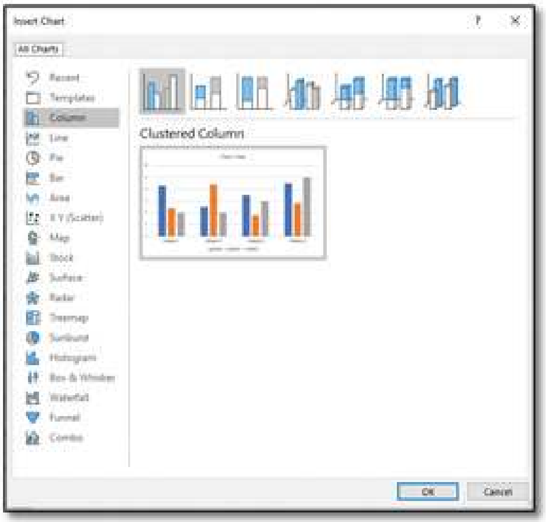


Gambar 3.2 Muncul Sheet seperti pada Microsoft Excel
Sumber: Dokumen Kemendikbud, 2021

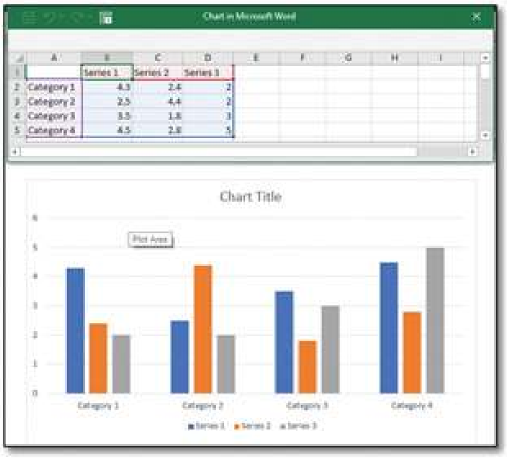
Karena praktis, kita sering melakukan *Copy-Paste*. *Copy-Paste* perlu dilakukan dengan cermat jika hasilnya masih harus diedit/diubah. Seringkali, terjadi *Copy-Paste* yang menghasilkan teks yang salah karena penulis lupa mengedit

2. Fitur Integrasi Konten pada Aplikasi

Integrasi konten pada bagian ini dilakukan dengan menggunakan fitur integrasi yang tersedia di aplikasi. Kita dapat menggunakan menu pada salah satu aplikasi yang secara otomatis membuka aplikasi lainnya. Misalnya, pada Microsoft Word, ketika dipilih menu *Insert Chart*, Microsoft Excel secara otomatis akan terbuka. Tentu, kalian juga membawa diagram ke MS PowerPoint dan melakukan hal yang sama. Gambar 3.3 dan Gambar 3.4 menunjukkan contoh menu *Insert Chart* yang dapat menampilkan sheet yang dapat diisi data untuk membuat Diagram.



Gambar 3.3 *Insert Chart* pada MS Word
Sumber: Dokumen Kemendikbud, 2021



Gambar 3.4 Muncul Sheet seperti pada Microsoft Excel
Sumber: Dokumen Kemendikbud, 2021

3. Object Linking & Embedding

Pada *Object Linking* (untuk Microsoft Office), objek yang berasal dari aplikasi perkantoran lain dapat disisipkan ke aplikasi perkantoran yang sedang dibuat. Perubahan pada data sumber akan mengakibatkan perubahan pada objek yang dihubungkan dengan fasilitas ini pada aplikasi lainnya.



Gambar 3.5 Ilustrasi *Object Linking* Antardua Dokumen
Sumber: Dokumen Kemendikbud, 2021

Pertanyaan Pemahaman

Jawablah pertanyaan di bawah ini dalam Buku Catatan, dan jangan lupa mencatat kegiatan dalam Jurnal.

1. Tahukah kalian, mengapa potongan teks, tabel atau gambar dapat di-copy/paste dan dibawa antaraplikasi?
2. Dalam Object Linking, jika sumber diedit, otomatis semua objek yang terhubung akan berubah. Menurut kalian, mengapa ini terjadi? Mana yang lebih menguntungkan, melakukan copy/paste atau object Linking?
3. Kelak, konsep ini akan kalian pelajari dalam pemrograman! Kedua konsep ini, yaitu (1) menyalin dan membawa salinannya, atau (2) hanya mengacu tanpa membawa objek yang disalin merupakan dua konsep yang penting dalam informatika. Konsep yang kedua akan menjadi dasar untuk belajar "pointer".

Ayo Lakukan

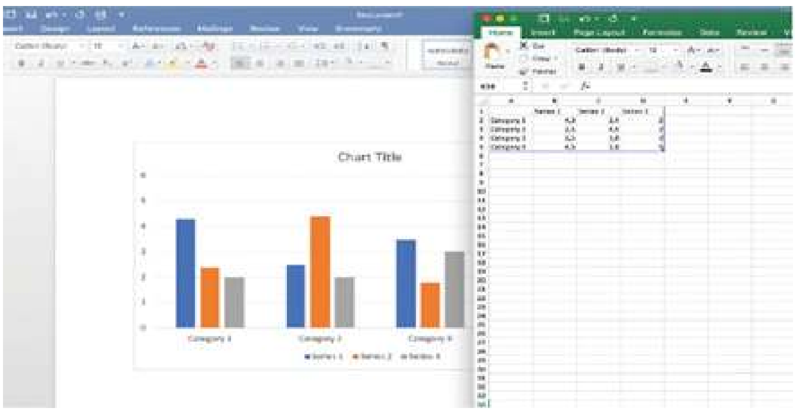
Aktivitas Individu

Aktivitas TIK-K10-01 Integrasi Word dan Excel

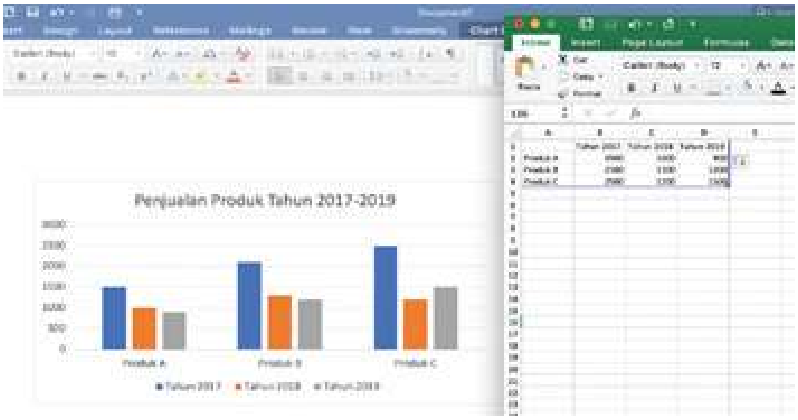
Seringkali, kalian perlu untuk membuat laporan yang mengandung gambar dan diagram (chart). MS Word menyediakan editor gambar tetap terbatas dan tidak direlasikan dengan data. Jika diagram berasal dari data yang diolah dan dibuat dengan MS Excel, kita perlu "memindahkan" diagram yang berasal dari data dan dibuat di MS Excel, menjadi potongan gambar pada MS Word. Kalau data berubah, kita harus mengulangi proses yang sama.

Langkah-Langkah Aktivitas (Tampilan yang disajikan menggunakan MS Office 2016):

1. Buka aplikasi MS Word.
2. Klik menu *Insert Chart.*
3. Pilih jenis diagram yang diinginkan. Maka, akan muncul tampilan diagram secara otomatis pada *Word* dan *Excel* yang memuat datanya, seperti pada gambar berikut.



4. Lakukan pengubahan pada data sesuai dengan yang diperlukan. Banyaknya baris maupun kolom dapat disesuaikan. Kemudian, Tutup MS Excel.
5. Ubahlah judul diagram dengan melakukan double click pada Chart Title. Tulis judul sesuai dengan yang diinginkan, seperti contoh berikut.



Pertanyaan Pemahaman

Jawablah pertanyaan di bawah ini dalam Buku Catatan, dan jangan lupa mencatat kegiatan dalam Jurnal!

1. Apakah kalian menyadari, bahwa dengan melakukan hal di atas, kalian mengaktifkan ("memanggil") MS Excel saat sedang memakai MS Word? Menurut kalian, apa yang terjadi? Mengapa bisa demikian?
2. Apa bedanya misalnya, dengan mengerjakan tabel dan gambar dengan menggunakan MS Excel, kemudian hasilnya dibawa dalam bentuk gambar (misalnya dengan "*sniper*" atau mengambil potongan layar menjadi gambar) sehingga tersimpan di Clipboard. Setelah itu, kalian membuka MS Word dan melakukan paste dalam dokumen MS Word.

Ayo Lakukan

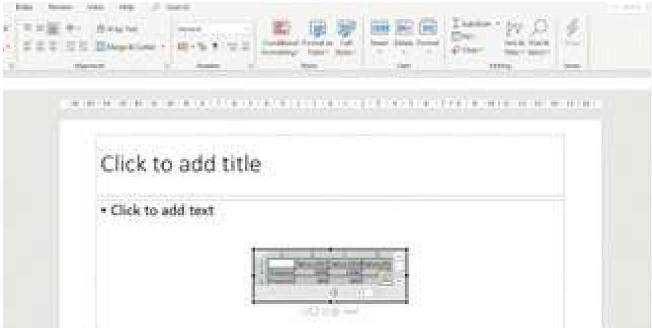
Aktivitas Individu

Aktivitas TIK-K10-02 Integrasi PowerPoint dan Excel

Dalam aktivitas ini, kalian akan membuat tabel excel sebagai bagian dari objek dalam sebuah slide PowerPoint.

Langkah-Langkah Aktivitas

1. Buka aplikasi MS PowerPoint.
2. Klik menu Insert Table Excel Spreadsheet. Perintah ini akan membuka Worksheet secara otomatis seperti berikut.



3. Buatlah tabel secukupnya. Meskipun yang terbuka hanya jendela kecil, tetapi fitur-fitur Excel, termasuk fungsi-fungsinya, tersedia secara lengkap seperti yang biasa terdapat pada aplikasi MS Excel. Peserta dapat mencoba fungsi-fungsi matematika yang biasa digunakan.
4. Apabila ingin update pada data, lakukan double click pada tabel, akan muncul kembali *worksheet*. Kemudian, lakukan perubahan sesuai yang diinginkan.

Ayo, Kita Praktik

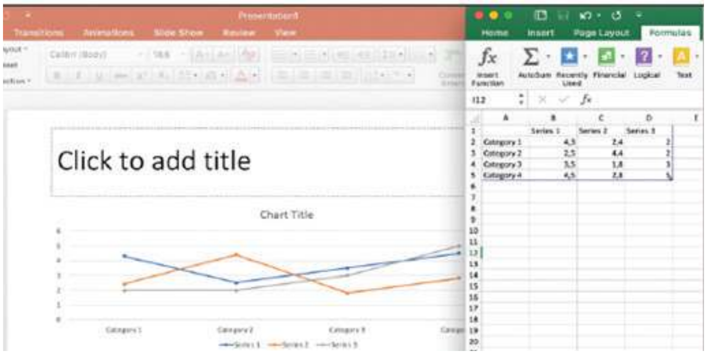
Aktivitas Individu

Aktivitas TIK-K10-03 Membuat Diagram Pada PowerPoint Dari Excel

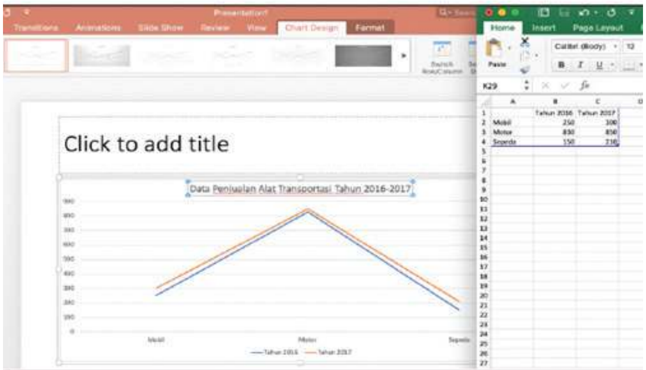
Kalian harus menyiapkan presentasi dari data hasil pengamatan. Karena data pengamatannya tidak banyak, lebih praktis untuk langsung mengetikkan data dan membuat diagram langsung pada slides.

Langkah-Langkah Aktivitas

1. Buka aplikasi MS PowerPoint.
2. Klik menu Insert Chart.
3. Pilih jenis diagram yang diinginkan. Maka, akan muncul tampilan diagram secara otomatis pada MS PowerPoint dan tampilan worksheet Excel yang memuat datanya, seperti pada gambar di bawah



4. Lakukan pengubahan pada data sesuai dengan yang diperlukan. Banyaknya baris maupun kolom dapat disesuaikan pula.
5. Tutup *worksheet*.
6. Ubahlah judul diagram dengan melakukan *double click* pada Chart Title. Tulis judul sesuai dengan yang diinginkan.



Ayo Lakukan

Aktivitas Individu

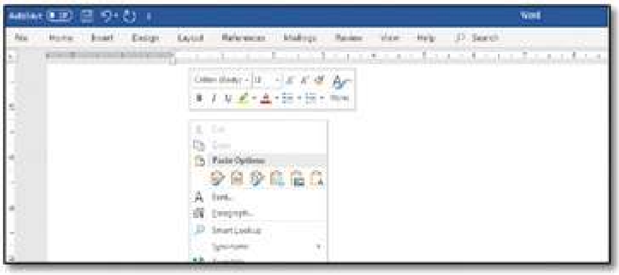
Aktivitas TIK-K10-04 OLE (*Object Linking & Embedding*) – Excel dan Word

Ada kalanya, kalian perlu memasukkan data yang telah dibuat pada MS Excel ke suatu laporan yang disusun dengan menggunakan MS Word. Dalam hal ini, data telah ada sehingga tinggal dimasukkan saja. Tentunya, salah satu cara yang paling cepat ialah menggunakan *copy-paste* biasa. Akan tetapi, dengan cara ini, data akan tersaji secara statis ialah data saat disalin sehingga jika terjadi perubahan pada data sumbernya di Excel, data pada Word tidak berubah. Hal ini bisa diatasi dengan menggunakan fasilitas *Object Linking and Embedding* (OLE).

Langkah-Langkah Aktivitas

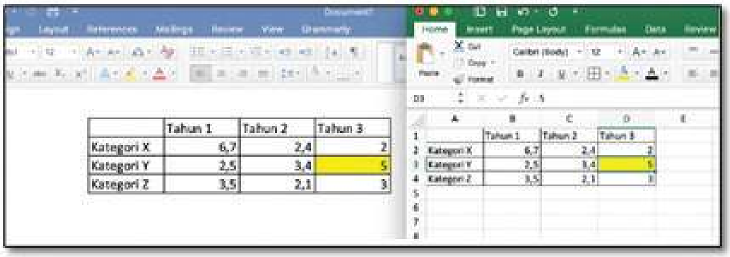
1. Buka aplikasi Excel.
2. Buatlah tabel pada Excel secukupnya.
3. Blok tabel yang sudah dibuat, kemudian copy tabel tersebut.
4. Buka Word.
5. Lakukan klik kanan.
6. Di bawah *Paste Options*, seperti terlihat pada gambar berikut, terdapat beberapa pilihan *paste*, pilih salah satu di antara *Link & Keep Source Formatting* (F) atau *Link & Use Destination Styles* (L). Perbedaan di antara keduanya hanyalah pada format tabelnya (misalnya *font*), yaitu

menggunakan format sumber dari Excel (untuk pilihan *Link & Keep Source Formatting*) atau menggunakan format pada Word (untuk pilihan *Link & Use Destination Styles*).



Selanjutnya, akan kita lakukan perubahan pada data sumber menggunakan perintah-perintah berikut ini.

7. Ubah data pada *file* Excel, misalnya dengan mengganti salah satu data pada tabel.
8. Perhatikan perubahannya pada Word.



Ayo Lakukan

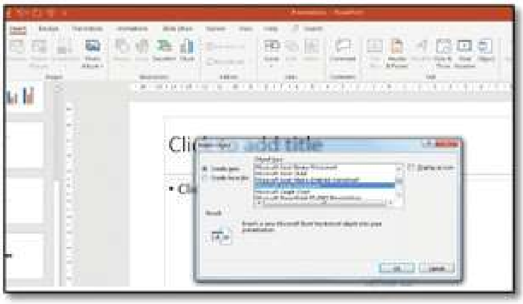
Aktivitas Individu

Aktivitas TIK-K10-05 OLE (Object Linking & Embedding) –Excel dan PowerPoint

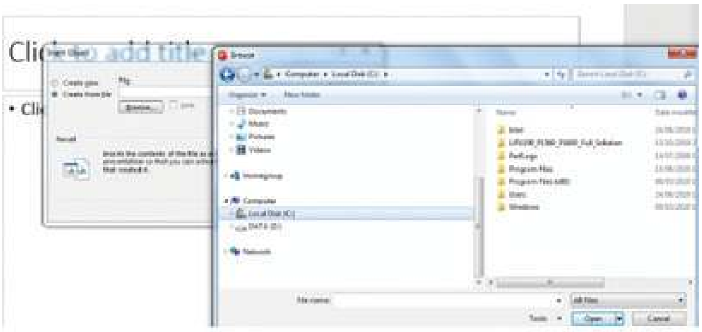
Pada aktivitas ini, kalian akan mencoba untuk menggunakan fitur OLE untuk menghubungkan tabel yang dibuat pada Excel ke *slide* pada PowerPoint.

Langkah-Langkah Aktivitas

1. Buka aplikasi Excel. Buatlah tabel pada Excel secukupnya. Simpan file Excel.
2. Buka MS PowerPoint.
3. Pilih Insert Object. Perintah ini akan membuka jendela seperti pada gambar berikut.



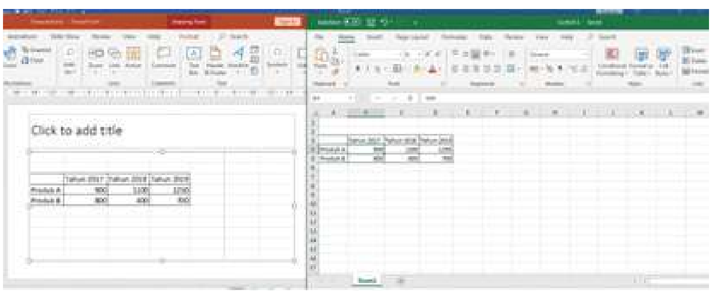
4. Pilih Create from file, kemudian pilih *file* Excel yang telah dibuat sebelumnya pada langkah 1-3.



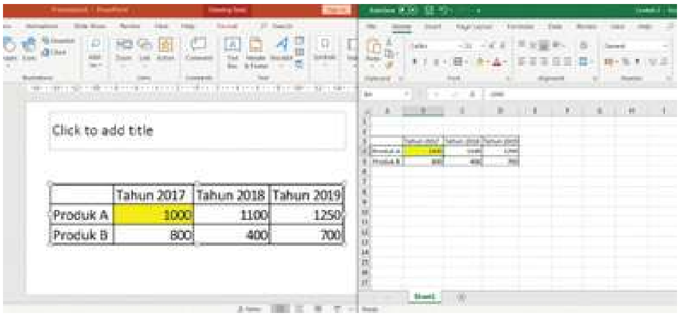
5. Beri tanda centang pada opsi *Link*, seperti gambar



6. Klik OK. Maka, pada PowerPoint akan tampil tabel sesuai dengan tabel yang telah dibuat pada *file* Excel. Tampilannya sebagai berikut.



7. Lakukan perubahan data pada *file* Excel, misalnya mengubah data seperti yang di-*highlight* kuning. Data pada *slide* PowerPoint juga ikut berubah, seperti ditunjukkan pada gambar sebagai berikut.



B. Fitur Lanjut Aplikasi Perkantoran

Banyak fitur lanjut dari aplikasi perkantoran yang bisa membantu kita dalam membuat laporan maupun menyajikan data. Pada bagian ini, kalian akan mencoba untuk mempraktikkan dua fitur lanjut aplikasi perkantoran, yaitu fitur *mail merge* pada Word dan fitur membuat video pada MS PowerPoint.

Ayo, Kita Praktik

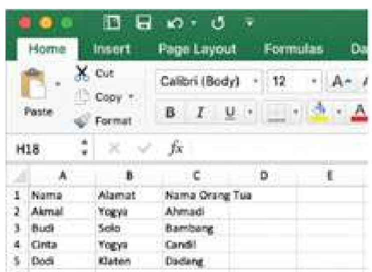
Aktivitas Individu

Aktivitas TIK-K10-06 Mail Merge

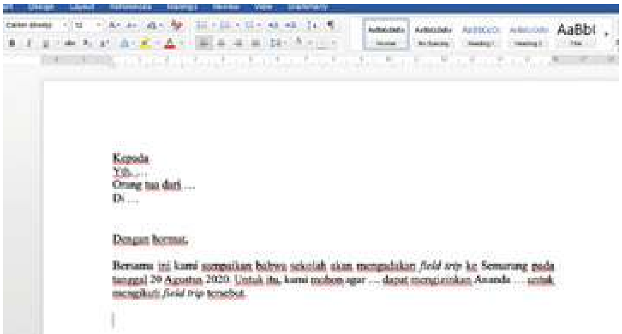
Mail Merge adalah fitur yang dapat kalian gunakan untuk menghasilkan surat, amplop, undangan, dan lain-lain secara berulang, tetapi terdapat beberapa komponen yang berbeda misalnya nama dan alamat yang dituju. Dalam hal ini, kita tidak perlu membuat dokumen sebanyak jumlah nama/tujuannya, tetapi hanya cukup menggunakan satu dokumen Word dan daftar nama/alamat yang dapat disimpan dalam daftar, basis data, atau spreadsheet.

Langkah-Langkah Aktivitas

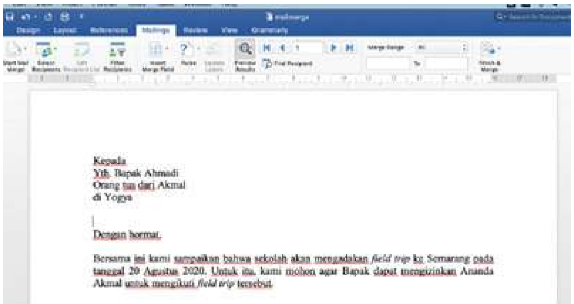
1. Buka Excel.
2. Buatlah tabel yang berisi data nama siswa, alamat, nama orang tua, kurang lebih seperti contoh di bawah ini.



3. Simpanlah *file* Excel tersebut.
4. Buka Word.
5. Buatlah dokumen surat kurang lebih seperti pada contoh berikut ini.



6. Klik Mailings Select recipients → Use an existing list.
7. Carilah *file* excel yang telah dibuat sebelumnya pada langkah 1-3.
8. Klik Open → pilih *sheet* yang sesuai → klik OK.
9. Letakkan kursor pada sebelah kanan 'Yth.'.
10. Klik Insert Merge Field → pilih Hubungan.
11. Ulangi langkah 9 dan 10 untuk bagian-bagian lain dari dokumen surat sehingga diperoleh hasil seperti gambar berikut.
12. Untuk melihat hasilnya, klik *Preview Result*. Tampilan hasil seperti pada gambar berikut.
13. Gerakkan panah ke kiri dan ke kanan untuk melihat tampilan data lainnya. Data pada setiap baris akan ditampilkan satu per satu.



Ayo Lakukan

Aktivitas Individu

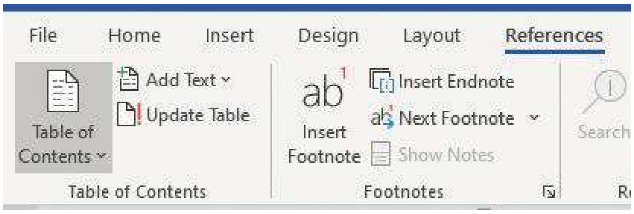
Aktivitas TIK-K10-07 Pembuatan Daftar Isi

Pada aktivitas ini, siswa mengeksplorasi sendiri sebuah tutorial MS Word untuk melakukan otomasi pembuatan Daftar Isi laporan.

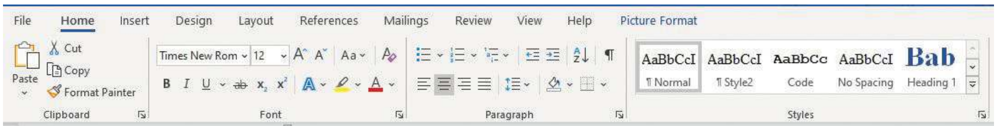
Pengantar

Pernahkah kalian melihat daftar isi sebuah buku? Daftar isi terdiri atas apa saja? Andaikata kalian sedang menulis laporan, alangkah susahnya jika harus mengetik judul bab satu per satu, kemudian mencatat nomor halamannya. Belum lagi jika teks berubah sehingga nomor halaman berubah. Adakah solusi?

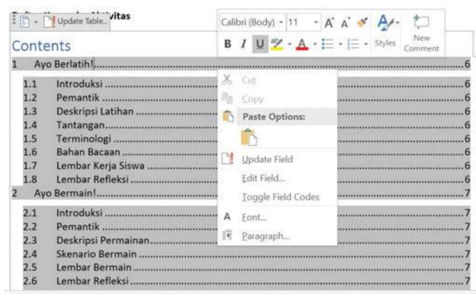
Salah satu fasilitas yang tersedia di Microsoft Word adalah pembuatan Daftar Isi (*Table of Content*). Fiturnya ada pada salah satu pilihan *Reference*.



MS Word dapat membuat daftar isi secara otomatis dengan merujuk pada dokumen yang telah diberi *Styles* pada menu *Home*.



Daftar isi dibuat dari teks yang diberi style Heading 1, Heading 2, Heading 3, dst. Jadi, kalian harus menandai judul bab dan subbab yang akan ditampilkan pada daftar isi dengan Style Heading 1, Heading 2, atau Heading 3. Tampilan tingkat kerincian pada daftar isi dapat disesuaikan sesuai dengan kebutuhan.



Gambar 3.6 Pembuatan Daftar Isi

Sumber: Kemendikbud, 2021

Langkah-Langkah Aktivitas

1. Posisikan kursor pada halaman di mana daftar isi akan dibuat (biasanya di awal).
2. Cari Pilihan "Table of Content", dan pilih format yang sesuai.
3. Daftar isi akan keluar secara otomatis.

Tutorial yang paling terpercaya tentunya diperoleh dari situs pengembang perangkat lunak, yaitu situs: https://support.microsoft.com/en-us/office/insert-a-table-of-contents-882e8564-0edb-435e-84b5-1d8552ccf0c0 untuk membuat daftar isi.

Jika kalian mengubah teks sehingga judul bab tergeser dan nomor halaman berubah, daftar isi harus diperbarui. Caranya mudah. Posisikan kursor pada daftar isi, klik kanan Mouse, akan muncul menu "Update Field". Pilih menu tersebut, dan Ooops... daftar isi secara otomatis berubah. Mudah, bukan?

Apa yang Harus Kalian Lakukan?

Ambillah sebuah laporan yang pernah kalian buat sebelumnya. Berilah *styles* pada judul bab dan subbab pada laporan tersebut. Pastikan judul bab pada setiap bab diberi *style* "Heading 1" dan judul bab pada setiap subbab adalah "Heading 2". Buatlah daftar isinya. Setelah itu, tambahkan bab dari *file* yang lain, dan tambahkan pada laporan kalian. Perbaharuilah daftar isi kalian. Mudah, bukan?

Tantangan:

Selain membuat daftar isi secara otomatis, kalian juga dapat mengacu secara otomatis nomor dan judul gambar, tabel atau potongan teks (misalnya akibat penomoran otomatis, saat menyisipkan gambar atau tabel, nomor bergeser. MS Word dapat menyesuaikan rujukan ke gambar, tabel, potongan teks atau rujukan lain secara otomatis. Eksplorasilah fitur pada menu "References", karena prinsipnya akan mirip dengan dengan pembuatan daftar isi. Identifikasi polanya, dan kalian akan dapat melakukannya sendiri. Kalian dapat mengeksplorasi semua menu MS Word, dan diskusikan dengan teman atau guru kalian. Kalian diharapkan dapat mengenali pola kesamaan dan perbedaannya!

Lembar Kerja Siswa

Setelah berlatih membuat daftar isi, isilah lembar kerja berikut.

Tugas	Mudah	Agak Sulit	Sulit	Sangat Sulit
Saya berhasil membuat Daftar Isi Laporan.				
Saya berhasil mengubah daftar isi secara otomatis setelah saya menambahkan satu bab.				
Saya berhasil mengubah format tampilan daftar isi.				
Ceritakanlah Pengalaman Kalian:				

Ayo Lakukan

Aktivitas Individu

Aktivitas TIK-K10-08 Membuat Video Presentasi dengan MS PowerPoint

MS PowerPoint dapat digunakan untuk membuat video presentasi. Versi yang baru dari aplikasi ini memiliki fitur untuk merekam audio dan video yang menyertai presentasi. Jangan lupa memakai aplikasi yang legal, bukan bajakan agar berfungsi dengan baik. Fasilitas ini bermanfaat untuk membuat video berupa presentasi yang disertai dengan narasi dan potongan video dari naratornya.

Apa yang Kalian Perlukan?

Komputer yang telah terpasang aplikasi presentasi MS PowerPoint yang menyediakan fitur perekaman video dan komputer mendukung perekam suara dan kamera.

Langkah-Langkah Aktivitas

1. Buatlah presentasi menggunakan PowerPoint.
2. Tampilkan tab Recording. Jika belum, dengan cara klik File → Options → Customize Ribbon → centang check box Recording → OK. Tab Recording akan muncul.
3. Klik tab Recording.
4. Klik Record Slide Show. Terdapat 2 pilihan: *Record from the beginning* atau *Record from Current Slide.*
5. Jika terdapat *webcam*, video dapat direkam dengan cara menekan tombol kamera kecil di bawah kanan yang muncul saat *Record slide show*.
6. Di bagian bawah layar, terdapat pen dan *highlighting tools* yang dapat digunakan untuk menandai layar (*drawing*) saat perekaman.
7. Untuk memulai perekaman, klik tombol *Record* di bagian kiri atas layar. Akan muncul *countdown timer*.
8. Perekaman dimulai, mulai berbicara dan menjelaskan.
9. Untuk berpindah slide, dapat digunakan panah pada keyboard atau pada layar.
10. Untuk mem-*pause*, klik tombol *Pause*.
11. Untuk menghentikan perekaman, tekan tombol *Stop.*
12. Jika terdapat kesalahan narasi pada salah satu *slide*, masuk ke *slide* tersebut, kemudian hapus perekaman pada slide tersebut menggunakan tombol *Clear* pada kanan atas, pilih *Clear Recording on Current Slid*e. Ini akan menghapus suara, *web footage*, maupun *drawing* pada slide. Untuk mengganti rekaman pada slide tersebut, tekan *Record* lagi, dan lakukan perekaman.
13. Jika ingin menghapus semua rekaman, tekan tombol *Clear* → *Clear Recording on All Slide*. Penghapusan ini permanen.
14. Setelah selesai melakukan perekaman, tekan tombol *Escape* (Esc) pada papan ketik atau X pada pojok kanan atas. Pada setiap *slide*, terdapat *recording living* pada bagian kiri bawah.
15. Presentasi dapat disimpan seperti itu saja, atau diekspor menjadi video MP4. Caranya, klik *Export to Video* → Pilih kualitas video → tekan tombol *Create a Video* → pilih di mana video akan disimpan.

Uji Kompetensi

Soal Uraian

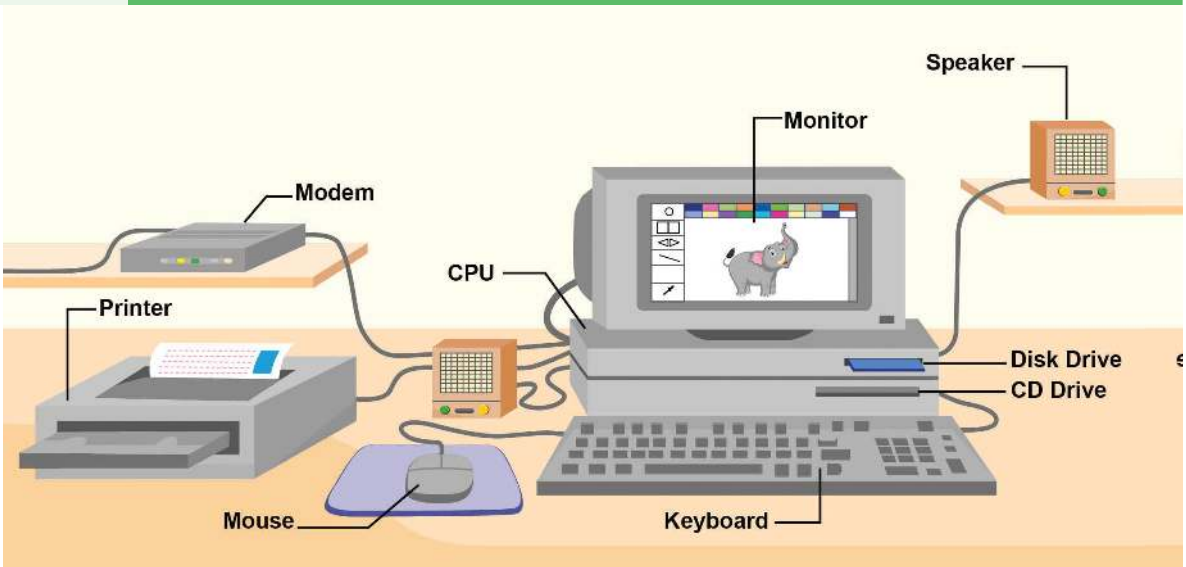
1. Sebagai pengurus OSIS, kalian diminta untuk membuat presentasi yang menyajikan grafik-grafik laporan keuangan yang datanya bersumber dari *file* Excel yang sudah ada sebelumnya. Karena data keuangan sangat penting, setiap ada perubahan data di *file* Excel, grafik pada *file* PowerPoint juga harus otomatis berubah. Tuliskan semua alternatif, kemudian jelaskan pilihan solusinya disertai penalaran mengapa solusi yang diusulkan paling optimal.
2. Jika kalian ingin membuat *file* presentasi yang berisi grafik-grafik dengan data baru, mekanisme apa yang bisa kalian lakukan untuk mengintegrasi Word dan Excel?
3. Jelaskan keuntungan dari OLE dibandingkan dengan cara integrasi lainnya. Jelaskan alasannya.
4. Kapan dan mengapa integrasi konten antaraplikasi perkantoran dibutuhkan? Jelaskan berbagai situasi yang membutuhkan integrasi konten yang pernah kalian alami.
5. Menurut kalian, mengapa video menjadi fitur lanjut yang bisa kalian gunakan dalam menyajikan presentasi dengan lebih baik yang ditambahkan pada PowerPoint?
6. Dalam waktu dekat, sekolah kalian akan mengadakan Pentas Seni Tahunan. Karena kalian diangkat sebagai sekretaris panitia, kalian harus membuat surat undangan perdana untuk rapat membahas Pentas Seni Tahunan ini dengan menggunakan para Ketua Kelas. Buatlah sebuah surat undangan dengan fasilitas Mail Merge.

Ayo Kita Renungkan

1. Pelajarilah menu dan fitur yang tersedia dalam aplikasi MS Word, MS Excel, dan MS PowerPoint. Buatlah tabulasi fiturnya. Simpulkan fitur apa saja yang terdapat dalam ketiga aplikasi tersebut.
2. Seberapa dalam kalian memahami "berpindah-pindah" dalam menggunakan aplikasi Office? Apakah kalian bingung?
3. Seberapa dalam kalian memahami makna "integrasi" di antara aplikasi Office? Apa yang diintegrasikan?
4. Apakah kalian dapat memberikan contoh permasalahan di sekitar yang dapat diselesaikan dengan mengintegrasikan aplikasi Office?
5. Pelajaran apa yang paling berkesan yang kalian dapatkan dari konsep integrasi antaraplikasi Office ini?
6. Selain beberapa aktivitas yang sudah dijelaskan dan kalian coba di atas, kira-kira apa lagi yang dapat kalian lakukan pada dalam penggunaan fitur lanjut pada aplikasi Office?

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Wahyono, Irya Wisnubhadra, Heni Pratiwi
ISBN : 978-602-244-506-7

Bab 4
Sistem Komputer



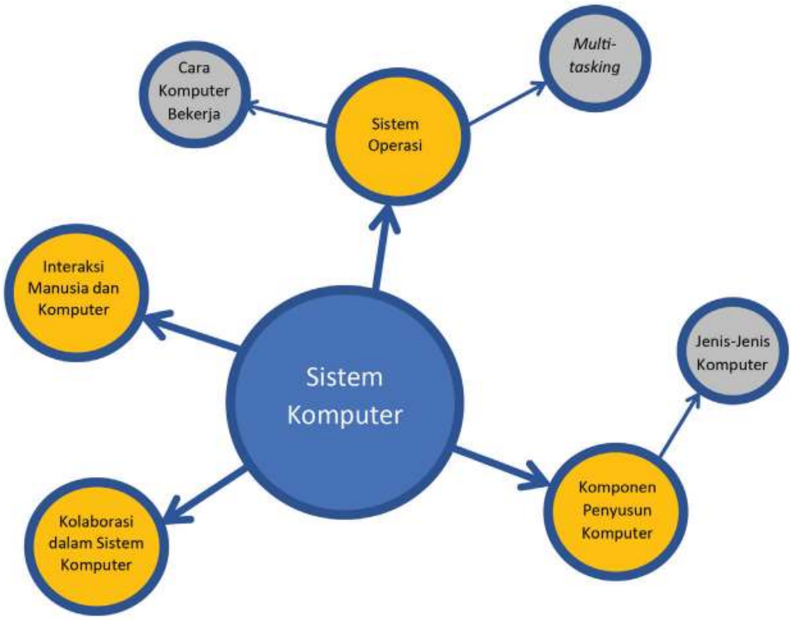
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu menjelaskan peran sistem operasi pada komputer, cara kerja komputer dalam memproses data, dan mekanisme internal yang terjadi pada interaksi antara perangkat keras, perangkat lunak, dan pengguna.

Pertanyaan Pemantik

Pada era digital saat ini, banyak kegiatan kita yang bergantung pada peralatan komputer. Tahukah kalian bahwa komputer bisa membantu kegiatan kita karena adanya sistem komputer yang dibentuk dari komponen-komponen? Bagaimana kerja sistem komputer?

Peta Konsep



Gambar 4.1 Peta Konsep Sistem Komputer
Sumber: Dokumen Kemendikbud, 2021

Apersepsi

Mungkin kalian pernah melihat bagaimana peranti elektronik bekerja, seperti mesin cuci, mesin pemotong rumput, dll. Komputer sebenarnya sama halnya dengan peranti lain, tetapi tampak lebih kompleks. Pekerjaan yang dilakukan oleh komputer dilakukan oleh banyak komponennya. Komputer juga memiliki arsitektur, yaitu tatanan komponen yang dirancang oleh pembuatnya. Pada materi ini, kalian akan belajar lebih dalam bagaimana sistem komputer bekerja.

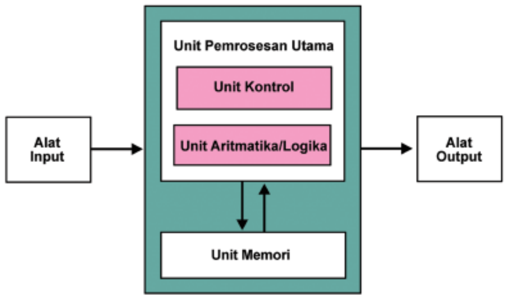
Kata Kunci

komponen komputer, CPU, ALU, perangkat lunak, perangkat keras, sistem operasi

Di jenjang SMP, kalian telah mengenal perangkat keras dan perangkat lunak. Perangkat tersebut dapat dijumpai sehari-hari di sekitar masyarakat yang sudah memanfaatkan TIK untuk membentuk sebuah sistem komputasi yang siap dimanfaatkan oleh pengguna. Selain itu, kalian juga telah memahami pentingnya sistem operasi sebagai perangkat lunak yang akan melayani perangkat keras dan perangkat lunak aplikasi yang dipakai oleh pengguna. Di kelas X, kalian diajak untuk menjenguk lebih dalam ke dalam "mesin" komputer. Gunanya untuk memahami komponen-komponen di dalam perangkat keras komputer secara lebih rinci, tetapi dalam bentuk abstraksi, dan bagaimana instruksi atau perintah yang ditulis sebagai program komputer dijalankan.

A. Komputer dan Komponen Penyusunnya

Secara umum, komputer adalah peralatan elektronik yang menerima masukan data, mengolah data, dan memberikan hasil keluaran dalam bentuk informasi, baik itu berupa gambar, teks, suara, maupun video. Secara sederhana, sebuah komputer menerima masukan dari peranti masukan, memproses masukan tersebut, dan menghasilkan output. Gambaran umum sebuah komputer tampak pada Gambar 4.2, di mana "data" (teks, suara, klik, sentuhan, atau lainnya) dari peranti masukan akan diproses oleh *Central Processing Unit* (CPU) untuk menghasilkan *output* berupa tampilan informasi, sinyal reaksi, dll. yang direpresentasikan melalui peranti keluaran.



Gambar 4.2 Diagram Blok Konseptual Komputer
Sumber: Dokumen Kemendikbud,2021

1. Sistem Komputer

Sistem Komputer terdiri atas beberapa bagian berikut.

1. Perangkat Keras (*Hardware*)
2. Perangkat Lunak (*Software*)
3. Pengguna

Semua komponen tersebut saling mendukung sehingga komputer dapat beroperasi. Perangkat keras komputer membutuhkan perangkat lunak agar komputer bisa dihidupkan dan difungsikan. Jika hardware yang tidak disertai *software*, komputer hanyalah sebuah mesin yang tidak berguna. Hal ini

dikarenakan *software* tercipta untuk menulis fungsionalitas pada komputer tersebut sehingga terciptalah sebuah komputer yang memiliki fungsi untuk digunakan. *Hardware* saja ibarat tubuh manusia tanpa jiwa.

a. Perangkat Keras (*Hardware*)

Perangkat keras komputer (*hardware*) adalah komponen fisik pada komputer yang dapat disentuh, dilihat atau dipindahkan. Contoh perangkat keras ialah mouse, harddisk, processor, RAM, printer, scanner dll.

b. Perangkat Lunak (*Software*)

Perangkat lunak komputer (*software*) tidak terlihat secara fisik, tetapi berfungsi dan dapat dioperasikan oleh pengguna melalui antarmuka yang disediakan. Fungsinya ialah untuk menjembatani pengguna dengan perangkat keras. Perangkat lunak adalah kode-kode program yang dibuat menggunakan bahasa pemrograman. Kode-kode tersebut merupakan kumpulan perintah atau instruksi untuk menjalankan tugas tertentu sesuai dengan keinginan pengguna, atau untuk mengendalikan kerja perangkat keras. Jika sebuah sistem komputer diibaratkan manusia, perangkat keras adalah "otak" dan perangkat lunak adalah "pikiran". Contoh perangkat lunak ialah sistem operasi, aplikasi (app), dll. Sistem operasi yang banyak dipakai saat ini ialah MS Windows, MacOS, dan Linux, sedangkan perangkat lunak lain yang menjadi sistem operasi ponsel pintar adialah Android.

c. Pengguna

Pengguna adalah orang yang menggunakan atau mengoperasikan komputer. Pemahaman lebih lanjut tentang pengguna, dapat ditemukan pada materi Dampak Sosial Informatika.

2. Jenis-Jenis Komputer

Berdasarkan ukurannya, komputer dibagi menjadi beberapa jenis antara lain seperti ditunjukkan pada penjelasan berikut.

Microcomputer (Komputer Mikro)

Komputer mikro merupakan komputer yang memiliki ukuran paling kecil dibandingkan dengan jenis komputer lainnya dan menggunakan microprocessor sebagai CPU atau unit pemrosesan utama. Contoh dari komputer mikro antara lain Ultrabook, permainan konsol, telepon pintar dan Tablet. Karena ukuran yang kecil dan harga yang lebih murah dibandingkan dengan jenis komputer lainnya, komputer mikro paling banyak digunakan dalam kehidupan sehari-hari.

Beberapa komputer bahkan dalam bentuk papan tunggal (*single board circuit*, SBC) yang berukuran kecil, misalnya yang populer ialah raspberry pi dan arduino.

Raspberry Pi, sering disingkat Raspi, ialah SBC yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris (https://www.raspberrypi.org/)



Gambar 4.3 Contoh Microcomputer

Arduino adalah *platform* elektronik open-source berdasarkan perangkat keras dan perangkat lunak yang mudah digunakan, ditujukan untuk membuat proyek interaktif. Papan arduino dapat membaca *input* dan menghasilkan sinyal output yang mengaktivasi motor, menyalakan LED, atau lainnya. Arduino dapat diprogram dengan mudah. Karena kemudahan dan harganya yang murah, arduino dapat ditemui mulai dari papan 8-bit sederhana hingga produk untuk aplikasi IoT, perangkat yang dapat dikenakan, pencetakan 3D, dan *embedded system.* (https://www.arduino.cc/)

Komputer Personal (PC, *Personal Computer*)

Komputer personal atau PC memiliki ukuran yang lebih besar dari komputer mikro dan memiliki kemampuan penyimpanan dan pengolahan data yang lebih besar dibandingkan dengan komputer mikro, dan dibuat untuk penggunaan personal. PC dapat berbentuk *desktop* PC (dirancang untuk ditaruh di meja), atau untuk dapat dijinjing dan dibawa-bawa (*laptop*).



Gambar 4.4 Contoh Personal Computer

Mini PC

Merupakan komputer "peralihan" dari komputer personal ke komputer mini yang dipakai di industri. Biasanya, dipakai untuk industri kecil atau personal untuk keperluan profesional atau industri kecil.



Gambar 4.5 Contoh Mini PC, Coffee Lake-U-based Bean Canyon Intel NUC8i5BEK2

Minicomputer



Gambar 4.6 Contoh Minicomputer PDP-8e Minicomputer System

Berbeda dengan komputer personal, komputer mini berukuran lebih besar, dan mempunyai kapasitas memori maupun pemroses yang lebih besar. Komputer mini dipakai menunjang kebutuhan pengolahan informasi perusahaan skala menengah. Saat ini, komputer mini kurang populer dan makin sedikit digunakan karena perusahaan lebih praktis untuk menyewa komputer di *cloud* yang memudahkan pemeliharaannya.

Komputer *Mainframe*



Gambar 4.7 Salah satu contoh Komputer Mainframe IBM z Systems z13.

Komputer *Mainframe* berukuran lebih besar dibandingkan dengan komputer dan biasanya digunakan oleh perusahaan-perusahaan besar sebagai *server* (peladen).

Supercomputer

Dibandingkan dengan komputer lainnya, *supercomputer* memiliki ukuran yang paling besar dan memiliki kapasitas pengolahan data dan kinerja yang paling kuat. Super Komputer memiliki kemampuan untuk melakukan triliunan perintah atau instruksi per detik yang dapat dihitung dalam FLOPS (*Floating Point Operation per Second*). Sama seperti minicomputer dan *mainframe*, pengguna super komputer biasanya ialah perusahaan atau organisasi besar, misalnya NASA yang menggunakannya dalam meluncurkan dan mengendalikan pesawat dan roket.



Gambar 4.8 Sierra/ATS-2 Super Computer

Catatan: selain "komputer" yang disebutkan di atas, beberapa gawai (*gadget*), yaitu perangkat elektronik kecil yang berfungsi khusus, ada yang termasuk "komputer" karena terdiri atas perangkat keras, sistem operasi dan perangkat lunak. Contohnya telepon pintar (*smartphone*) dan tablet, yang makin populer dan menjadi perlengkapan sehari-hari di era digital ini. Selain fungsi utamanya untuk berkomunikasi, ponsel pintar bahkan sudah menjadi "asisten" pribadi.



Gambar 4.9 Abstraksi Interaksi Perangkat Keras, Perangkat Lunak, dan Pengguna

Sumber: Dokumen Kemendikbud, 2021

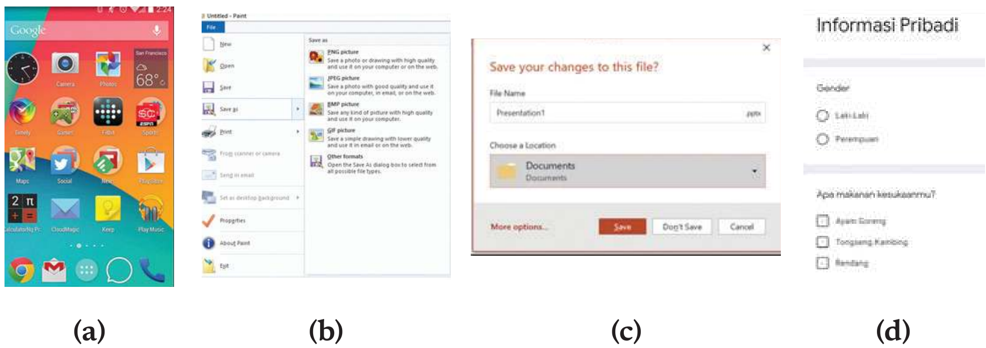
B. Interaksi Manusia dan Komputer

Sebuah sistem komputasi terdiri atas perangkat keras dan perangkat lunak yang saling berinteraksi. Interaksi antarmuka seringkali digunakan untuk menghubungkan perangkat satu dengan perangkat lainnya, mulai dari perangkat masukan dan keluaran yang dikendalikan oleh sistem operasi, dan kemudian dapat ditambahkan berbagai aplikasi yang dapat dipakai oleh pengguna.

Berbagai macam jenis interaksi dapat dilakukan oleh pengguna dengan sistem komputer, di antaranya seperti berikut.

1. Berbasis GUI (*Graphical User Interface*)

Merupakan antarmuka yang menggunakan menu grafis untuk memudahkan pengguna berinteraksi dengan komputer. GUI merupakan antarmuka pada sistem operasi komputer yang menggunakan menu grafis. Pengguna berinteraksi melalui ikon, menu, dialog dengan *button* dan *text box*, radio *button* (untuk satu pilihan), atau *checkbox* (untuk banyak Pilihan).

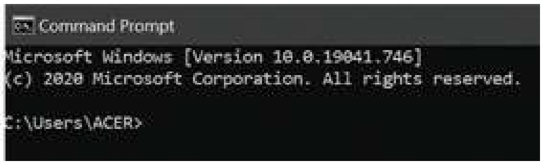


Gambar 4.10 Antarmuka Berbasis Grafis (a) Ikon pada ponsel, (b) Menu, (c) Dialog, textbox, dan button, (d) checkbox dan radio button

Sumber: Dokumen Kemendikbud, 2021

2. Antarmuka Berbasis Perintah (*Command Line Interface/CLI*)

Sistem operasi berbasis CLI merupakan tipe antarmuka melalui *text-terminal.* Pengguna menjalankan perintah dan program di sistem operasi tersebut dengan cara mengetikkan baris-baris tertentu.



Gambar 4.11 Contoh Antarmuka Berbasis Perintah (CLI)
Sumber: Dokumen Kemendikbud, 2021

3. Melalui Suara (*Audio*)

Antarmuka menggunakan suara memungkinkan pengguna mengucapkan sesuatu dan hasilnya akan direkam, dalam bentuk format audio. Antarmuka ini hanya dimungkinkan jika perangkat keras menyediakan perekam suara dan melalui aplikasi.

4. Melalui Gambar (*Video*)

Antarmuka menggunakan gambar hanya dimungkinkan jika perangkat keras menyediakan kamera. Kamera akan merekam gambar dan melalui aplikasi akan menyimpan gambar dalam format video.

5. Melalui Berbagai Piranti Masukan Lainnya

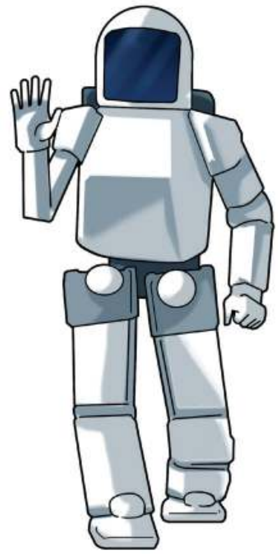
Selain melalui perangkat lunak, pengguna dapat berinteraksi langsung ke komputer melalui piranti masukan, seperti yang dijelaskan sebelumnya (*keyboard, joystick, mouse, touchpad*, layar sentuh, *keyboard virtual*, dll).

C. Kolaborasi dalam Sistem Komputer

Dari semua uraian di atas, terlihat bahwa di dalam sebuah sistem komputer, perangkat keras berkolaborasi dengan perangkat lunak aplikasi atau piranti lain melalui sistem operasi. Sistem Operasi pun berkolaborasi dengan aplikasi untuk berinteraksi dengan pengguna. Kolaborasi itu menghasilkan sebuah sistem komputasi yang akan bermanfaat bagi pengguna. Komputer tunggal seperti seorang manusia, yang di dalam tubuhnya beroperasi sistem-sistem yang berfungsi sesuai peran dari setiap organ tubuh. Pikiran manusia akan menggerakkan anggota tubuh seperti tangan dan kaki karena adanya sistem syaraf. Indra penglihat, peraba, perasa akan memberikan signal kepada sistem syaraf untuk membuat manusia bereaksi, bergerak, dan melakukan tindakan. Inilah yang menjadi dasar manusia menciptakan robot-robot yang bertindak dan berperilaku sesuai program yang disimpan dalam robot tersebut.

Pada sistem-sistem otonom, sistem komputasi tidak hanya berkomunikasi dengan manusia, tetapi juga berinteraksi dengan alam sekitarnya. Perekam gambar, suara, dan signal lainnya akan dapat mengirimkan data ke sistem untuk diolah. Inilah dasar dari sistem IoT (*Internet of Things*) di mana di sekeliling manusia dipasang perangkat-perangkat yang mampu menangkap data dan mengirimkan ke komputer. Dalam sebuah komputer, komponen-komponennya saling berhubungan. Demikian juga antara satu komputer dan komputer lainnya akan berhubungan melalui jaringan, mulai dari jaringan lokal sampai jaringan global internet. Berikut penjelasan ringkas kolaborasi tersebut.

Bahwa dalam sistem komputer, terjadi interaksi dan kolaborasi. Ini dapat dijumpai dalam kehidupan sehari-hari, seperti software yang digunakan di sebuah toko, warnet, kantor, rumah sakit, dan sebagainya. Seperti ketika melakukan instalasi program aplikasi penggajian (*payroll*) di komputer atau laptop kantor yang datanya masih perlu diolah dengan pengolah lembar kerja. Contoh lainnya ialah ketika membeli *laptop* baru dan akan membuat laporan di dalamnya, perlu menginstal sistem operasi terlebih dulu, sebelum aplikasi pengolah kata. Hal ini menunjukkan bahwa kedua *software* di atas saling bekerja sama satu sama lainnya, sebuah *software* bergantung pada *software* lainnya.



Gambar 4.12 Valkyrie, Humanoid Buatan NASA
Sumber: Dokumen Kemendikbud, 2021

Salah satu contoh dari interaksi *hardware* dengan sesama *hardware* ialah saat kita mencolokkan sebuah perangkat keras dengan perangkat keras lain. Misalnya, mencolokkan *flashdisk* atau *mouse* ke komputer. Saat kita mengambil *photo* menggunakan *webcam* yang ada pada komputer dan kemudian hasilnya akan dicetak menggunakan printer, apakah ini interaksi antarperangkat keras? Pada kasus ini, *webcam* akan mengirimkan gambar ke komputer, dan gambar yang akan dicetak ke *printer* dibuka oleh aplikasi lain. Interaksi *webcam* dan memori komputer dilakukan melalui sistem operasi yang dijalankan pada komputer. Interaksi antarperangkat keras ini membutuhkan perangkat lunak sebagai perantara.

Interaksi antarmuka *hardware* dan *software* terlihat sedikit lebih rumit karena *software* harus dirancang dan dibuat agar dapat mengenali atau dikenali oleh *hardware*. Contohnya, sebelum menggunakan *printer*, biasanya pengguna

akan memasang sebuah program yang disebut *driver* agar komputer dapat untuk mengenali *printer*. *Driver* pada komputer adalah komponen system *software* yang berfungsi sebagai perangkat komunikasi antara sistem operasi dan *hardware.* Driver menggunakan kode biner untuk memerintah *hardware* melakukan perintah yang diberikan oleh sistem operasi dan mengambil data yang dikirimkan oleh *hardware.* Begitu juga selanjutnya, sistem operasi memberikan data ke *printer* agar *printer* bekerja mencetak dokumen. Program yang akan dipasang ialah termasuk *software,* dan *printer* yang dihubungkan dengan komputer atau *laptop* sebagai *hardware.* Dapat disimpulkan, telah terjadi hubungan yang memerlukan antarmuka antara *software,* komputer atau *laptop*, dan pengguna.

Contoh lainnya ialah ketika akan bermain musik. Kita dapat menggunakan alat tambahan menyerupai *flashdisk* (Makey Makey, misalnya pada modul https://brainsintheclouds.eu/?page_id=159) yang dapat ditancapkan ke komputer, kemudian dengan sebuah kabel, dapat disambungkan ke pisang, wortel, jeruk, alumunium, dan sebagainya. Untuk memainkannya, dapat membuka situs web atau aplikasi tertentu, lalu menyentuh barang yang disambungkan dengan kabel, dan musik pun dapat dimainkan. Ketika akan bermain aplikasi permainan menggunakan ponsel, pasti membutuhkan *software* aplikasi permainan tersebut sehingga terjadi interaksi antarmuka antara ponsel sebagai hardware dan *software* aplikasi permainan.

D. Sistem Operasi

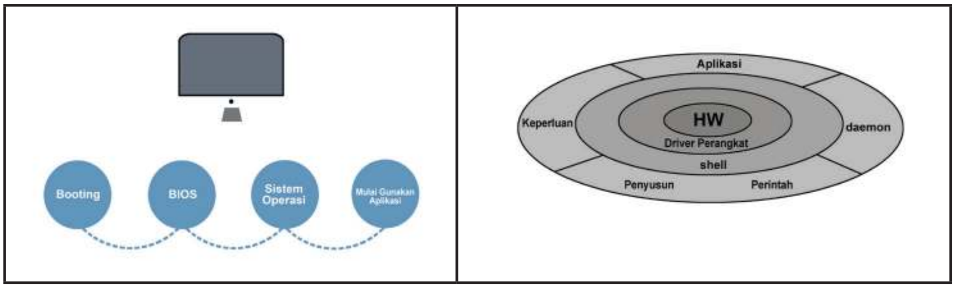
Sistem operasi (OS) adalah perangkat lunak sistem yang mengelola perangkat keras komputer, sumber daya perangkat lunak, dan menyediakan layanan umum untuk program komputer. Sistem operasi berfungsi untuk mengendalikan, mengontrol atau memberikan koneksi antarperangkat keras komputer.



Gambar 4.13 Beberapa Contoh Sistem Operasi
Sumber: Dokumen Kemendikbud, 2021

Contoh dari sistem operasi ialah MS Windows, Linux, MacOS, Android, iOS, dan sebagainya. Sistem operasi berfungsi untuk mengelola proses, memori utama, mengelola file, mengelola penyimpanan sekunder, serta melakukan proteksi dan keamanan.

Setelah komputer yang sudah diinstal sistem operasi MS Windows dihidupkan dan proses *booting* (penyalaan awal komputer) selesai, secara otomatis, sistem akan mengarahkan ke login MS Windows. Namun, jika baru untuk pertama kalinya dan tidak menggunakan password, tampilan login MS Windows akan dilewati dan berlanjut ke tampilan desktop.



Gambar 4.14 Alur Sistem Operasi pada Komputer dan "Lapisan Bawang" Sistem Komputer
Sumber: Dokumen Kemendikbud, 2021

Pada saat sebuah komputer dalam keadaan hidup, Sistem Operasi (SO) hidup di atas perangkat keras dan mengendalikan perangkat keras serta melayani perangkat lunak. Saat sebuah komputer dalam keadaan mati, tentu Sistem Operasi mati. Lalu, bagaimana Sistem Operasi mulai dihidupkan saat perangkat keras mulai dihidupkan? Pernahkah kalian memikirkan "c*hicken and egg problem*" ini?

Semua program komputer termasuk Sistem Operasi dimuat dalam CPU saat dijalankan (hal ini akan dijelaskan pada bagian berikutnya). Pada saat sebuah komputer dihidupkan, CPU-nya belum mengandung program. Proses menyalakan komputer dan membuat Sistem Operasi mulai bekerja disebut *booting. Booting* adalah urutan *startup* (membangunkan) Sistem Operasi komputer saat dinyalakan. Pada saat *booting*, semua komponen perangkat keras dalam komputer dibangunkan. Ada 5 langkah penting saat *booting*: daya (*power*) dihidupkan, POST (*Power On Self Test*), memuat BIOS (*Basic Input Output System*), memuat sistem operasi, kemudian perangkat keras mengalihkan kontrol ke Sistem Operasi.

Fungsi Sistem Operasi secara lebih rinci dapat dijelaskan sebagai berikut.

1. Menyediakan antarmuka ke pengguna untuk berinteraksi dengan komputer. Tanpa antarmuka yang mudah digunakan, komputer akan sulit untuk digunakan.
2. Mengendalikan *input* dan *output*.
3. Mengelola perangkat keras dan periferal. Pada saat pengguna mengetik di *keyboard*, aplikasi yang dipakai akan bekerja sama dengan Sistem Operasi untuk menangani *signal* dari *keyboard* untuk diproses. Misalnya, saat menggunakan pemroses kata, *mouse* yang digerakkan

kursornya akan terlihat di layar, huruf-huruf yang diketikkan dari *keyboard* (perangkat *input*) akan muncul di layar monitor (perangkat keluaran). Pada modul AP, saat kalian membaca sebuah nilai variabel, aplikasi menunggu sampai pengguna mengetikkan data yang dibaca, dan program akan memroses data yang diketikkan sesuai dengan algoritma program kalian. Sistem Operasi akan menjadi perantara program aplikasi untuk berinteraksi dengan *keyboard* atau perangkat *input/output* lainnya.

4. Mengelola pemuatan perangkat lunak dan menjalankan perangkat lunak. SO menjalankan program aplikasi agar dapat berfungsi dengan baik. Program aplikasi dirancang untuk dapat berfungsi di atas sistem operasi tertentu. Hal ini memungkinkan program aplikasi mendapat keuntungan dari koneksi dan fungsi perangkat keras yang tersedia melalui sistem operasi karena aplikasi yang sama dapat dijalankan pada perangkat keras yang berbeda-beda.
5. Mengelola *file (copy, save, sort, delete)*. Sistem Operasi mengelola sistem pengarsipan *file* yang dibuat untuk mengatur *file* dan direktori (*folder*). Adanya sistem *file* ini menyediakan cara yang konsisten untuk menyimpan dan mengambil data. Sistem Operasi juga bertanggung jawab atas penamaan, penyortiran, penghapusan, pemindahan, dan penyalinan *file/folder* (atas permintaan pengguna). Sistem Operasi mengelola sebuah Tabel pencarian yang digunakan untuk menghubungkan nama *file/folder* dan lokasi penyimpanan. Pengelolaan *file* dan *folder* ini dilakukan SO seperti kalian mengelola Buku Kerja Siswa.
6. Menangani interupsi dan kesalahan (*error*). Beberapa program dapat disimpan dalam RAM pada waktu bersamaan, tetapi prosesor hanya dapat memproses satu per satu. Melalui penggunaan sinyal interupsi yang reguler, Sistem Operasi dapat mengatur permintaan penggunaan prosesor dan menentukan prioritas program yang dijalankan. Hal ini membuat seakan-akan CPU menangani lebih dari satu program sekaligus (*multitasking*). Sistem Operasi juga dapat mendeteksi kesalahan yang biasanya ditangani tanpa perlu mengingatkan pengguna.
7. Mengelola prosesor. Sistem Operasi juga mengelola CPU. Saat sebuah program aplikasi diaktifkan, Sistem Operasi memuat program tersebut ke memori (RAM) dan CPU akan mengeksekusinya (penjelasan detail mengenai eksekusi program akan dijelaskan dalam pembahasan tentang mesin konseptual sederhana). CPU kemudian diinstruksi oleh Sistem Operasi untuk menjalankan program. Jadi, SO

bekerja sama dengan CPU untuk menjalankan sebuah program. Pada saat *multitasking* (menjalankan banyak program sekaligus), Sistem Operasi mengatur penggunaan waktu prosesor dengan menggilir pekerjaan. Bagian Sistem Operasi yang mengelola penggiliran CPU disebut *scheduler* (penjadwal).

8. Mengelola memori. Sistem operasi bertanggung jawab untuk mentransfer program ke memori dan dari memori. Sistem Operasi melacak penggunaan memori dan memutuskan berapa banyak yang harus diberikan untuk setiap program. Sistem Operasi juga memutuskan apa yang terjadi jika tidak ada cukup memori.
9. Mengelola keamanan. Sistem Operasi bertanggung jawab atas pembuatan akun pengguna dan kata sandi serta penerapannya. Sistem Operasi juga dilengkapi dengan banyak program utilitas, termasuk *firewall*.
10. Menangani komunikasi dengan jaringan.

Dari semua fungsi di atas, kalian dapat membayangkan, bahwa sebuah SO pun terdiri atas komponen-komponen dengan tugasnya masing-masing. Setiap komponen akan menjalankan tugas sesuai fungsinya yang spesifik. Pada kegiatan berikut ini, kalian akan lebih memahami salah satu fungsi Sistem Operasi dengan menjalankan tugas *scheduler*, yang mirip dengan tugas-tugas manusia menangani pekerjaan. Seperti halnya kalian berbagi peran dalam mengerjakan sebuah tugas, Sistem Operasi terdiri atas komponen-komponen dengan peran masing-masing yang disebutkan di atas, yang secara keseluruhan berfungsi sebagai Sistem Operasi.

1. Multitasking

Kalian tentu pernah melakukan "multitasking", yaitu mengerjakan beberapa pekerjaan sekaligus, misalnya sambil merajut, nonton TV, bahkan sesekali menengok HP. Sambil menyapu, kalian mendengarkan musik bahkan ikut bernyanyi. Manusia mempunyai kemampuan *multitasking*, walaupun untuk beberapa kondisi, perlu dilakukan dengan hati-hati, misalnya sangat berbahaya menonton video sambil menyetir mobil. *Multitasking* yang tidak dikendalikan dengan baik, belum tentu menambah efisiensi dan hasilnya belum tentu baik. Misalnya, belajar sambil menonton sepak bola dapat memecah perhatian sehingga kalian tidak belajar dengan baik. Ketika kalian sedang belajar sambil mendengarkan lagu, kalian berhenti ketika ibu memanggil untuk makan malam (ini yang disebut interupsi).

Sebuah komputer yang sedang melakukan multitasking, misalnya saat penggunanya sedang menjalankan aplikasi pengolah kata, aplikasi pengolah lembar kerja, aplikasi presentasi, dan aplikasi Paint untuk menggambar.

Pengguna memindahkan sepotong teks dari satu aplikasi ke lainnya lewat *clipboard* yang sebetulnya juga sebuah "aplikasi". Diam-diam, jam yang tertulis di pojok layar juga sedang bekerja. Jika sedang terhubung ke jaringan, tiba-tiba email masuk dan komputer menampilkan pesan. Komputer dan SO sangat "sibuk", bukan?



(Sumber: Dokumen Kemdikbud, 2021)

Seperti dijelaskan di atas, salah satu fungsi sistem operasi ialah menangani *multitasking*. Sistem Operasi tidak menangani *multitasking* seperti manusia karena komputer hanya mempunyai satu prosesor, dan prosesor itu yang menjalankan program (lihat aktivitas mesin super konseptual tentang bagaimana CPU menjalankan program dengan langkah sangat rinci).

Sistem Operasi dapat melakukan *multitasking* dengan menjalankan algoritma "*round robin*" (RR). Ya, sebuah algoritma karena Sistem Operasi adalah sebuah program juga. Prinsip dari algoritma penjadwalan *round robin* dijelaskan sebagai berikut.

Round-Robin (RR) ialah salah satu algoritma yang digunakan oleh penjadwal proses (*process scheduler*) dalam sebuah sistem operasi. Pada algoritma RR, ditentukan suatu slot waktu (*time slice*) yang akan dialokasi ke setiap proses dalam porsi yang sama dan dalam urutan melingkar, menangani semua proses tanpa prioritas. Penjadwalan RR sederhana, dan mudah diterapkan. Penjadwalan RR dapat diterapkan pada masalah penjadwalan lainnya, seperti penjadwalan paket data di jaringan komputer. Nama algoritma ini berasal dari prinsip *round-robin*, di mana setiap orang mengambil bagian yang sama dari sesuatu secara bergantian.

Agar proses dikerjakan secara adil, penjadwal RR memberikan setiap pekerjaan slot waktu atau penyisihan waktu CPU, dan menginterupsi pekerjaan belum selesaikan saat itu. Pekerjaan dilanjutkan saat slot waktu berikutnya ditetapkan bagi proses itu. Jika proses selesai atau mengubah statusnya menjadi menunggu selama slot waktu yang diberikan, penjadwal memilih proses pertama dalam antrean siap untuk dieksekusi. Dengan tidak ada banyak pekerjaan yang dilakukan, atau jika slot waktu relatif besar terhadap ukuran pekerjaan, proses yang menghasilkan pekerjaan besar akan lebih banyak dikerjakan daripada proses lainnya.

Misalnya, jika slot waktu ialah 100 milidetik (*mili second/ms*), dan job1 membutuhkan total waktu 250 ms untuk menyelesaikannya, penjadwal RR akan menangguhkan pekerjaan setelah 100 ms dan memberikan waktu pada pekerjaan lain di CPU. Setelah pekerjaan lain memiliki bagian yang sama (masing-masing 100 ms), job1 akan mendapatkan alokasi waktu CPU lain dan siklus akan berulang. Proses ini berlanjut hingga pekerjaan selesai dan tidak membutuhkan waktu lagi di CPU.

Job1	membutuhkan 250 ms untuk dapat diselesaikan, dengan slot 100 ms
Alokasi pertama	100 ms.
Alokasi kedua	100 ms.
Alokasi ke-3	100 ms tetapi job1 selesai dan diakhiri [ada 50 ms.

Jadi, waktu CPU untuk job1 = 250 ms.

Ada dua pendekatan algoritma untuk menyelesaikan RR *scheduler.*

Algoritma Pertama (dengan algoritma ini, CPU tidak pernah berhenti)

1. Selama periode satu slot waktu (kuantum): jika ada job selesai, hapus dari antrean, ambil berikutnya.
2. Di akhir satu kuantum: antrekan kembali, ambil giliran berikutnya

Algoritma Kedua

1. Selama Periode Satu Kuantum: Jika Ada Job Selesai, Hapus Dari Antrean, Tunggu Sampai Akhir Kuantum.
2. Di Akhir satu kuantum: antrekan kembali, ambil giliran berikutnya.

Tabel berikut menunjukkan waktu kedatangan dan waktu eksekusi dari beberapa proses, dengan slot 100 ms dan eksekusi dari proses-proses tersebut.

Tabel 4.1 Waktu Kedatangan dan Eksekusi Proses

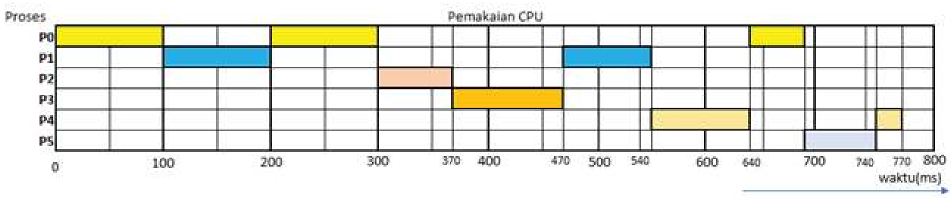
Kuantum = 100 ms

Proses	Waktu Kedatangan	Waktu Eksekusi (ms)
P0	0	250
P1	50	170
P2	120	70
P3	170	100
P4	200	130
P5	350	50
	Total	770

SIMULASI PENJADWALAN PROSES

Waktu	Antrian							Selesai	Deskripsi Kejadian
0	P0								P0 datang , diproses
50	P0	P1							P1 datang masuk antrian
100	P1	P0							P1 diproses, P0 diantrikan kembali
120	P1	P0	P2						P2 datang masuk antrian
170	P1	P0	P2	P3					P3 datang masuk antrian
200	P0	P2	P3	P1	P4				P0 diproses, P4 datang masuk antrian
300	P2	P3	P1	P4	P0				P2 diproses, P0 diantrikan kembali
350	P2	P3	P1	P4	P0	P5			P5 datang masuk antrian
370	P3	P1	P4	P0	P5			P2	P2 selesai, P3 diproses
470	P1	P4	P0	P5				P3	P3 selesai, P1 diproses
540	P4	P0	P5					P1	P1 selesai, P4 diproses
640	P0	P5	P4						P0 diproses, P4 diantrikan kembali
690	P5	P4						P0	P0 selesai, P5 diproses
740	P4							P5	P5 selesai, P4 diproses
770								P4	P4 selesai

(a)



(b)

Gambar 4.15 (a) Simulasi Penjadwalan Proses, (b) Simulasi CPU
Sumber: Dokumen Kemendikbud,2021

Nah, perancang sistem dapat mengatur kuantum yang optimal jika perilaku pekerjaan (*job*) yang harus ditangani oleh komputer diketahui. Ini membutuhkan berpikir komputasional. Pada komputer personal, hal ini tidak penting. Namun, dalam sebuah sistem komputer *mainframe* atau sistem komputasi dengan banyak mesin yang melayani banyak pengguna dengan karakteristik masing-masing, mengoptimasi pelaksanaan pekerjaan menjadi sangat penting.

Interpretasi Diagram!

Jawablah pertanyaan berikut ini.

1. Mengacu ke simulasi eksekusi tersebut, pekerjaan (proses) mana yang selesai terlebih dahulu dan mana yang selesai paling akhir?
2. Kesimpulan apa saja yang kalian dapatkan?
3. Dari dua algoritma yang dijelaskan di atas, algoritma mana yang dipilih?

Kalian juga dapat membuat diagram eksekusi yang lebih visual dengan sticky notes, untuk mempermudah interpretasinya.

Ayo Lakukan

Aktivitas Kelompok

Aktivitas SK-K10-01-U Simulasi Multitasking (*Unplugged*)

Pada aktivitas ini, kalian berlatih untuk menerapkan sistem *multitasking* dengan menggunakan simulasi dan permainan peran (*role play*).

Apa yang kalian perlukan?

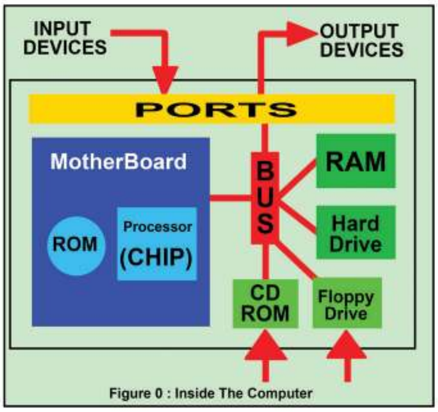
1. Papan tulis atau *whiteboard*
2. Spidol *boardmarker* berwarna atau kapur berwarna
3. *Sticky note* berwarna

Langkah-Langkah:

1. Setiap kelompok akan berbagi.
   a. Berperan sebagai "*scheduler*" (penjadwal) yang merupakan bagian dari SO. *Scheduler* mengelola permintaan layanan dan antrean, mencatatnya dalam lembar catatannya. Setiap job yang datang akan memberi tahu sebelumnya berapa lama waktu eksekusi yang dibutuhkannya.
   b. CPU (yang menjalankan job, menahannya selama mendapat giliran) dan melepaskannya saat slot waktu habis.
   c. Bus yang akan membawa dan mengembalikan "job" ke antrean yang dikelola *scheduler* berikut statusnya (selesai, atau masih perlu berapa lama lagi).
2. Guru akan memberikan sebuah tabel yang isinya waktu kedatangan beberapa *job* untuk dilayani oleh SO.
3. Simulasikan pelaksanaan pekerjaan mulai kedatangan job yang pertama sampai semua *job* selesai dikerjakan.

2. Cara Komputer Bekerja

Sekarang, kita akan mempelajari mekanisme kerja pemroses karena pemroses pun memiliki banyak komponen yang harus melakukan interaksi satu sama lain. Secara ringkas, Gambar 4.16 menggambarkan diagram kotak arsitektur sederhana sebuah komputer di mana pemroses (kotak besar) menerima masukan dari piranti masukan dan menghasilkan



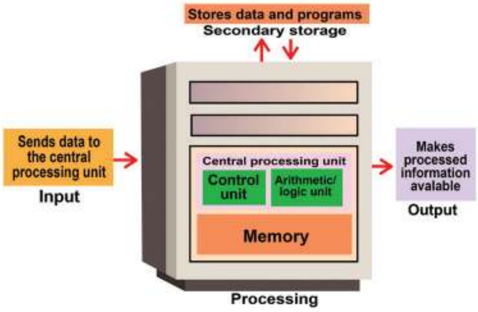
Gambar 4.16 Komponen Internal pada Komputer
Sumber: Dokumen Kemendikbud, 2021

keluaran melalui *port*. Komponen pada pemroses (kotak besar) terdiri atas *processor* (CPU) yang berupa chip, ROM, RAM, Hard drive, CD ROM, dan *floppy drive* (sekarang tidak populer). CD ROM dan *Floppy drive* ialah alat yang terhubung dengan media penyimpan eksternal, yaitu (CD dan *floppy disk*). Setiap komponen tersebut terhubung dengan BUS.

Berikut ini akan dijelaskan mengenai CPU, Bus, Controller, dan memori utama.

Central Processing Unit (CPU)

Sekarang, kita akan melihat apa yang dikerjakan oleh pemroses sebuah komputer yang tak terlihat, yaitu sebuah pusat pengontrol yang mengonversi data input menjadi output. Pengontrol ini disebut CPU (*Central Processing Unit*). CPU adalah sebuah sirkuit elektronik yang sangat kompleks, yang bertugas mengeksekusi instruksi yang disimpan sebagai program. Semua komputer, mulai yang kecil seperti ponsel pintar sampai yang besar dan canggih, harus mempunyai CPU. Seperti ditunjukkan pada Gambar 4.17, CPU terdiri atas dua bagian, yaitu CU (*control unit*) dan ALU (*Arithmetic/Logic Unit*). Setiap bagian mempunyai fungsinya masing-masing.



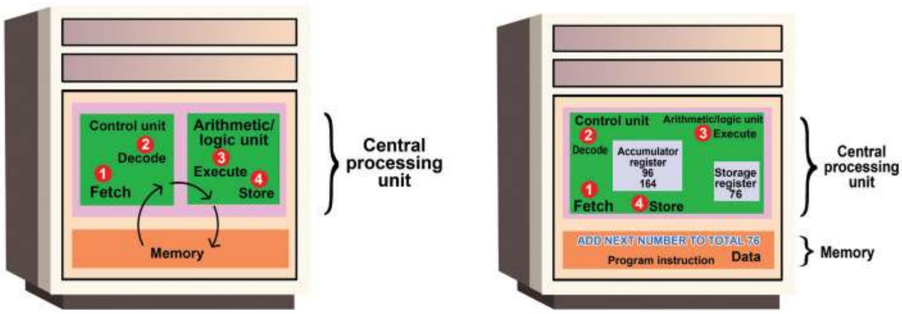
Gambar 4.17 Central Processing Unit
Sumber: Dokumen Kemendikbud, 2021

Sebelum kita membahas CU dan ALU secara rinci, kita perlu mempelajari penyimpanan data dan hubungannya dengan CPU. Komputer mempunyai dua jenis tempat penyimpanan: primer (*primary storage*) dan sekunder (secondary storage). CPU berinteraksi dengan tempat penyimpanan primer atau memori utama (*main memory*), mengacu ke data dan instruksi yang disimpan di dalamnya. Dengan alasan ini, memori dibahas terkait CPU, walaupun sebetulnya secara teknis, memori merupakan bagian terpisah dari CPU.

Ingat, bahwa memori utama komputer menyimpan data sementara saja, saat komputer mengeksekusi sebuah program. Memori sekunder menyimpan data permanen atau semi permanen dalam media lain, yaitu media magnetik atau optik. Disket dan CD-ROM yang sering kita lihat sebagai bagian dari PC adalah peranti sekunder, sama halnya dengan *hard disk*. Memori sekunder akan dibahas pada bagian berikutnya.

a. Bagaimana CPU Menjalankan Instruksi Program

Mari, kita periksa cara unit pengolah pusat (CPU), dalam hubungannya dengan memori, menjalankan program komputer. Kita akan melihat bagaimana hanya satu instruksi dalam program yang dijalankan. Faktanya, kebanyakan komputer saat ini hanya dapat menjalankan satu instruksi pada satu waktu, meskipun mereka menjalaninya dengan sangat cepat. Banyak komputer pribadi dapat menjalankan instruksi dalam waktu kurang dari sepersejuta detik. Komputer yang dikenal sebagai superkomputer dapat menjalankan instruksi dalam waktu kurang dari sepermiliar detik.



Gambar 4.18 Siklus Mesin pada Komputer Sebelum dan Setelah Bekerja
Sumber: Dokumen Kemendikbud, 2021

Sebelum instruksi dapat dieksekusi, instruksi program dan data harus ditempatkan ke dalam memori dari perangkat input atau perangkat penyimpanan sekunder. Seperti yang ditunjukkan Gambar 4.18 (sebelah kiri), setelah data dan instruksi yang diperlukan berada dalam memori, unit pemrosesan pusat melakukan empat langkah berikut untuk setiap instruksi.

1. Unit kontrol mengambil (mendapat) instruksi dari memori.
2. Unit kontrol menerjemahkan instruksi (memutuskan apa artinya) dan memerintahkan agar data yang diperlukan dipindahkan dari memori ke ALU (unit aritmatika/logika). Dua langkah pertama ini bersama-sama disebut waktu instruksi, atau waktu-I.
3. Unit aritmatika/logika menjalankan instruksi aritmatika atau logika. Artinya, ALU diberikan kendali dan melakukan operasi aktual pada data.
4. Unit aritmatika/logika menyimpan hasil operasi ini dalam memori atau register. Langkah 3 dan 4 bersama-sama disebut waktu eksekusi, atau waktu-E.

Unit kontrol selanjutnya memerintahkan memori untuk mengirimkan hasilnya ke perangkat keluaran atau perangkat penyimpanan sekunder. Kombinasi waktu-I dan waktu-E disebut siklus mesin. Gambar 4.18 (kanan) menunjukkan instruksi yang melalui siklus mesin.

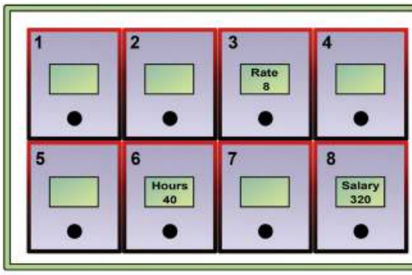
CPU memiliki *clock* internal yang menghasilkan detak (*pulse*) dengan kecepatan tetap untuk menyinkronkan semua operasi komputer. Sebuah instruksi siklus mesin tunggal dapat terdiri atas sejumlah besar sub-instruksi, yang masing-masing harus mengambil setidaknya satu siklus *clock*. Setiap jenis CPU dirancang untuk memahami sekumpulan instruksi tertentu yang disebut *instruction set.* Sama seperti, ada banyak bahasa berbeda yang dipahami orang, setiap jenis CPU memiliki instruction set yang dimengertinya. Oleh karena itu, CPU yang digunakan untuk PC merk Compaq tidak dapat memahami kumpulan instruksi dari CPU lain (misalnya Macintosh).

Pada memori, dapat tersimpan instruksi program dan juga data. Bagaimana *control unit* bisa membedakannya?

Lokasi dalam memori untuk setiap instruksi dan setiap bagian data diidentifikasi oleh sebuah alamat. Artinya, setiap lokasi memiliki nomor alamat, seperti *locker* (kotak penyimpan) di perpustakaan atau di sekolah. Seperti kotak penyimpan memiliki memiliki nomor yang tetap, isi kotak penyimpan bisa berbeda di suatu waktu, bisa berisi tas, bisa berisi buku, atau berisi *tumbler* tempat minum.

Seperti hal kotak penyimpan, memori bisa berisi instruksi atau data. Instruksi lama bisa diganti dengan instruksi baru, demikian juga data lama bisa diganti dengan data baru, tetapi memori tetap memiliki alamat yang sama. Tidak seperti kotak penyimpan, lokasi memori hanya dapat menampung sejumlah data dalam ukuran byte.

Gambar 4.19 menunjukkan bagaimana program memanipulasi data dalam memori. Sebuah program penggajian, misalnya, dapat memberikan instruksi untuk meletakkan data tarif (gaji-per jam) di lokasi kotak 3 dan jumlah jam kerja di lokasi kotak 6. Untuk menghitung gaji karyawan, instruksi untuk komputer, yaitu mengalikan data di lokasi kotak 3 dengan data di lokasi kotak 6 dan pindahkan hasilnya ke lokasi kotak 8. Pemilihan lokasi dapat dilakukan di mana saja yang belum digunakan. Pemrogram yang menggunakan bahasa pemrograman tidak perlu tahu nomor alamat mesin yang sebenarnya, karena setiap alamat data disebut dengan nama yang menjadi alamat simbolis. Dalam contoh ini, nama alamat simbolis adalah Tarif, Jam, dan Gaji.



Gambar 4.19 Memory Address seperti Kotak Surat

sumber: Dokumen Kemendikbud, 2021

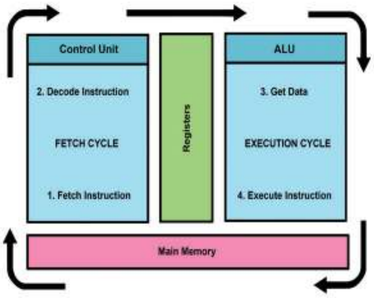
b. Mesin Konseptual Sederhana

Tahukah kalian, bahwa kita bisa membuat abstraksi sebuah mesin komputer dengan menciptakan sebuah mesin konseptual sederhana?

Kalian diberi sebuah mesin konseptual sederhana ciptaan Mr. ALGO dan menyimulasikan cara kerjanya, yang seperti cara kerja sebuah komputer, tetapi dengan lebih sederhana. Pada bagian ini, kalian akan memahami bahwa instruksi program dalam bahasa yang lebih dekat ke manusia harus diterjemahkan menjadi instruksi dalam bahasa mesin untuk dapat dijalankan. Ingat bahwa dengan struktur komputer yang terdiri atas input, output, memori dan CPU, komputer hanya dapat membaca data dari perangkat masukan, menulis data ke perangkat keluaran, menyalin data dari CPU ke memori atau dari memori ke CPU, dan melakukan perhitungan aritmatika dan logika. Dengan kemampuan tersebut, kita dapat menuliskan program komputer yang beragam dan luar biasa daya gunanya!

Program komputer terdiri atas sekumpulan instruksi. Instruksi yang dijalankan oleh mesin harus dikenali oleh CPU yang disebut bahasa mesin. Bagaimana sebuah komputer menjalankan sebuah program dalam bahasa mesin? Komputer akan menjalankan (mengeksekusi) perhitungan dengan langkah yang disebut "*fetch execute cycle*" (siklus ambil dan jalankan). *Fetch execute cycle* ialah operasi yang paling mendasar dalam komputer, yang juga disebut "*fetch decode execute cycle*". Selama *fetch execute cycle*, mesin komputer akan mengambil instruksi dari memori dan menjalankan instruksi tersebut sesuai dengan jenis instruksinya. Mengapa disebut *fetch execute cycle*? Siklus untuk *fetching, decoding,* dan *executing* sebuah instruksi ini akan diulang-ulang oleh CPU selama mesin komputer hidup!

Gambaran siklusnya ditunjukkan dalam Gambar 4.20 yang menunjukkan lebih detail bagian-bagian dari CPU.



Gambar 4.20 Siklus Ambil dan Jalankan (Fetch Execute Cycle)

Sumber: Dokumen Kemendikbud, 2021

1. CU akan mengambil instruksi (Fetch).
2. CU akan menerjemahkan instruksi tersebut harus melakukan apa, misalnya menyimpan data, menghitung, atau lainnya. Ingat, bahwa komputer hanya dapat melakukan perhitungan dan menyimpan/mengambil, atau mengirimkan data ke *input/output device.*

3. ALU akan mengambil data yang diperlukan untuk menjalankan instruksi, dan data yang sedang diproses disimpan dalam Register.
4. ALU menjalankan instruksi.

Berikut ini gambaran Memori dan Register dari sebuah mesin ciptaan Mr. ALGO tersebut. Alamat pada mesin ini dinyatakan dalam kode Heksadesimal. Memori dibagi-bagi menjadi kotak-kotak. Setiap kotak mempunyai alamat dan dapat berisi data. Misalnya, mesin ciptaan kalian memiliki kapasitas memori untuk menampung 4 data dengan alamat AAA1 s.d AAA4 dan CPU mempunyai 2 register dengan alamat REG1 dan REG2 yang ilustrasikan seperti pada gambar di bawah ini. Sebetulnya, semua data akan disimpan dalam bentuk biner, seperti yang telah dipelajari di jenjang SMP. Namun demikian, untuk kemudahan membaca, data semua ilustrasi, data tetap dituliskan dalam besaran desimal.

MEMORI					REGISTER	
AAA1	AAA2	AAA3	AAA4		REG1	REG2

Sekarang, kalian akan mensimulasi *Fetch instruction cycle* tersebut untuk menjalankan instruksi perhitungan aritmatika sederhana sebagai berikut yang semula ditulis dalam bahasa yang lebih dipahami manusia (bahasa tingkat tinggi).

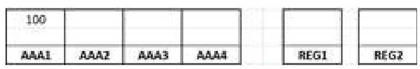
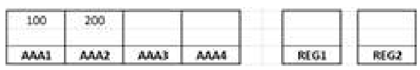
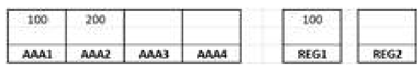
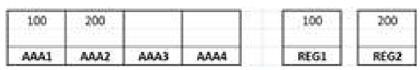
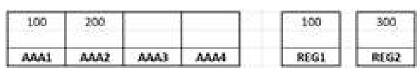
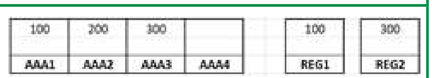
```
X=200
Y=100
Jumlah= X + Y
PRINT Jumlah
```

Mesin Konseptual Sederhana akan mengeksekusi perintah tersebut dalam beberapa langkah karena data disimpan dalam memori (disimpan dalam variabel X dan Y), sedangkan proses perhitungan penjumlahan harus dilakukan oleh ALU yang merupakan bagian dari CPU. Misalnya, nilai dari variabel X disimpan dalam alamat AAA1 dan nilai dari variabel Y disimpan dalam AAA2, dan nilai variabel Jumlah akan disimpan dalam alamat AAA4. Data harus dibawa ke Register untuk dijumlahkan.

Mesin Konseptual Sederhana menjalankan beberapa instruksi bahasa mesin sebagai berikut untuk menjalankan program di atas.

SIMPAN 100 AAA1
SIMPAN 200 AAA2
SALIN AAA1 REG1
SALIN AAA2 REG2
TAMBAH REG1 REG2
SALIN REG2 AAA3
PRINT AAA3

Eksekusi dari perintah-perintah tersebut secara berturut-turut ditunjukkan dengan ilustrasi sebagai berikut.

Instruksi	Penjelasan	Peta Memori dan CPU
SIMPAN 100 AAA1	Menyimpan data 100 ke dalam memori dengan alamat AAA1.	
SIMPAN 200 AAA2	Menyimpan data 200 ke dalam memori dengan alamat AAA2.	
SALIN AAA1 REG1	Menyalin data dari memori dengan alamat AAA1 ke register dengan alamat REG1.	
SALIN AAA2 REG2	Menyalin data dari memori dengan alamat AAA2 ke register dengan alamat REG2.	
TAMBAH REG1 REG2	Menambahkan data pada alamat REG1 dan data pada alamat REG2 dan mesin akan menyimpan hasilnya pada alamat REG2. *) Baca Penjelasan di bawah tabel.	
SALIN REG2 AAA3	Menyalin data dari register REG2 ke memori AAA3.	
PRINT AAA3	Mencetak data yang ada pada memori AAA3 ke layar monitor mesin untuk ditampilkan.	**300**

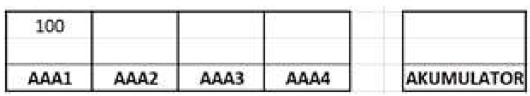
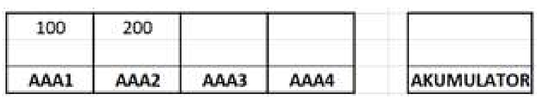
Mesin Konseptual Sederhana tersebut hanya mempunyai 2 register (REG1 dan REG2), dan dirancang agar jika melakukan operasi aritmatika dua buah bilangan, hasilnya disimpan pada register yang menyimpan operan ke-2.

Mungkin saja, ada orang yang menciptakan mesin konseptual lain yang mempunyai 3 register. Seandainya, mesin mempunyai 3 register, bisa saja hasil penjumlahan disimpan pada register ke-3. Jumlah register dan bagaimana operasi dilakukan, serta hasil operasi aritmatika disimpan, itu ditentukan oleh penciptanya, merupakan spesifikasi mesin.

Kita juga dapat menciptakan mesin konseptual yang sangat sederhana, yang hanya mempunyai 1 register untuk menampung data, yang biasanya disebut AKUMULATOR karena semua operan dan hasil perhitungan disimpan di register tunggal. Perintah TAMBAH dilakukan dengan menambahkan data yang diambil dari memori ke satu-satunya register itu. Perintah Mesin misalnya:

INSTRUKSI	HASIL
LOAD <alamat>	Data yang disimpan pada alamat memori disimpan ke AKUMULATOR.
TAMBAH <alamat>	Data yang disimpan pada AKUMULATOR ditambah dengan data yang diambil dari alamat memori.
KURANG <alamat>	Data yang disimpan pada AKUMULATOR dikurangi dengan data yang diambil dari alamat memori.
KALI <alamat>	Data yang disimpan pada pada AKUMULATOR dikalikan dengan data yang diambil dari alamat memori.
BAGI <alamat>	Data yang disimpan pada AKUMULATOR dikurangi dengan data yang diambil dari alamat memori.
OUTPUT	Data AKUMULATOR dikirim ke perangkat keluaran.
INPUT <data><alamat>	Alamat diisi dengan data.

Untuk menambahkan 100+200, instruksi yang dilakukan oleh mesin dengan satu AKUMULATOR yang hanya mempunyai 4 alamat memori AAA1, AAA2, AAA3 dan AAA4 diberikan sebagai berikut.

INSTRUKSI	ISI MEMORI dan CPU (Akumulator)
INPUT 100 AAA1	
INPUT 200 AAA2	

INSTRUKSI	ISI MEMORI dan CPU (Akumulator)
LOAD AAA1	AAA1: 100, AAA2: 200, AAA3: , AAA4: AKUMULATOR: 100
TAMBAH AAA2	AAA1: 100, AAA2: 200, AAA3: , AAA4: AKUMULATOR: 300
OUTPUT	**300**

Jika kalian perhatikan, instruksi-instruksi di atas merupakan rangkaian perintah yang dilakukan untuk menghitung penjumlahan 100+200. Instruksi yang diberikan bergantung pada spesifikasi mesinnya, dan mesin juga dapat mempunyai satu atau lebih register. Pencipta mesin yang akan merancang mesin-mesin komputer tersebut, dan mewujudkannya menjadi perangkat elektronik yang dapat berfungsi sesuai spesifikasi. Tentu saja, gambaran pelaksanaan instruksi di atas ialah penyederhanaan karena yang dilakukan oleh komputer sebenarnya jauh lebih rinci. Data 100 dan 200 dalam desimal pun akan direpresentasi dalam bentuk biner seperti yang kalian telah pelajari di tingkat SMP.

Unit pembelajaran ini memberikan gambaran bahwa ekspresi yang ditulis dalam bahasa yang dimengerti manusia harus diterjemahkan sangat detail agar dapat dieksekusi oleh mesin ciptaan kalian. Komputer yang selama ini kalian gunakan sebenarnya mirip dengan mesin konseptual sederhana tersebut, hanya kecepatan melaksanakan instruksi sangat cepat sekali.

Ayo Lakukan

Aktivitas Individu

Aktivitas Sk-K10-02-U Simulasi Eksekusi Perintah Dalam Mesin Ciptaan Mr. ALGO

Pada aktivitas ini, kalian akan bermain untuk memahami lebih detail lagi tentang bagaimana data disimpan dan diproses dalam sebuah komputer konseptual.

Apa yang kalian perlukan?

1. Sticky notes
2. Spidol
3. Kertas berukuran A0

Apa yang perlu kalian lakukan?

Guru kalian akan membuat kelompok yang terdiri atas 3-4 siswa.

Mr. ALGO menciptakan mesin konseptual sederhana dengan 5 alamat memori AEB1, AEB2, AEB3, AEB4, AEB5. CPU-nya memiliki 3 Register REG1, REG2 dan REG3. Spesifikasi instruksi-instruksinya sebagai berikut.

Instruksi MESIN	Penjelasan
SIMPAN <data><alamat>	Menyimpan data ke alamat memori yang dituju.
SALIN <alamat1><alamat2>	Menyalin data dari alamat1 untuk kemudian disimpan pada alamat2. Alamat dapat berupa alamat memori atau register.
TAMBAH REG1 REG2	Melakukan penjumlahan data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
KALI REG1 REG2	Melakukan perkalian data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
KURANG REG1 REG2	Melakukan pengurangan data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
BAGI REG1 REG2	Melakukan pembagian data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
PRINT <alamat>	Mencetak data yang ada pada alamat memori untuk ditampilkan pada monitor.

Tugas kalian ialah menerjemahkan operasi matematika berikut ke dalam instruksi-instruksi supaya bisa dieksekusi oleh mesin ciptaan Mr. ALGO tersebut.

1. $3 + 8 \times 9$
2. $(1 + 2) \times (8 - 5)$
3. $2 \times 10 - 8 + 3$
4. $3 + 16/2$

Dengan menggunakan *sticky note*, jangan lupa untuk menggambarkan peta dari memori dan CPU pada setiap instruksi yang dieksekusi oleh mesin kalian ke dalam kertas berukuran A0.

Ayo Kita Renungkan

Suatu hari, jika akan menjadi ahli perancang komputer, kalian akan berkenalan dengan *chips*, atau komponen elektronik yang membentuk mesin ciptaan. Tentu, harus membuat bahasa yang dipahami oleh komponen-komponen itu dengan memanfaatkan *interface* yang tersedia. Menantang, bukan?

Uji Kompetensi

1. Diberikan dua mesin ciptaan Mr Algo yang lain. Setiap spesifikasinya sebagai berikut ini:

Spesifikasi Mesin-1: operasi aritmatika 2 operan hasilnya disimpan pada register yang menyimpan operan pertama. Misalnya, TAMBAH REG1 REG2 hasilnya disimpan pada REG1.

Spesifikasi Mesin-2: operasi aritmatika 2 operan hasilnya disimpan pada register yang menyimpan operan kedua. Misalnya, TAMBAH REG1 REG2 hasilnya disimpan pada REG2.

Isilah sel yang kosong dengan isi memori AAA1, AAA2, AAA3, dan isi register REG1, REG2, REG3.

Urutan Instruksi	Mesin 1	Mesin 2
SIMPAN 8 AAA1		
SIMPAN 9 AAA2		
SIMPAN 10 AAA3		
SALIN AAA1 REG1		
SALIN AAA2 REG2		
SALIN AAA3 REG3		
TAMBAH REG1 REG2		
TAMBAH REG2 REG3		
KALI REG2 REG3		
KALI REG1 REG3		
SALIN REG1 AAA3		
SALIN REG2 AAA2		
SALIN REG3 AAA1		

Dengan mengacu ke mesin yang hanya mempunyai 1 akumulator dan 4 alamat memori AAA1, AAA2, AAA3 dan AAA4, tuliskan urutan instruksi dan gambaran isi memori dan register, untuk melakukan operasi perhitungan:

$(1 + 4) \times (8 - 5)/5$

dengan menggunakan memori se-efisien mungkin.

Ayo Kita Renungkan

Kalian telah belajar tentang sistem operasi dan interaksi antara komputer dan perangkat lain. Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Bayangkan seberapa penting sistem operasi dalam interaksi ini.
2. Dari materi ini, apakah kalian telah mengerti alur kerja sistem operasi?
3. Jelaskan dengan kata-kata sendiri, interaksi antara perangkat keras, perangkat lunak, dan pengguna yang kalian ketahui.
4. Menurut kalian, interaksi seperti apa yang paling sering dilakukan ketika kalian menggunakan komputer untuk keperluan pembelajaran?

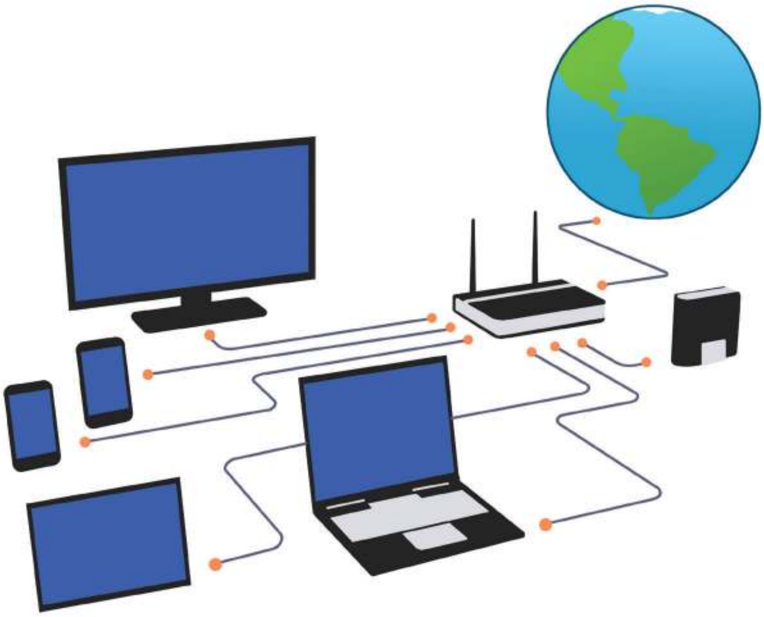
Ingin Tahu Lebih?

Jika kalian tertarik dengan materi ini dan ingin mendalaminya lebih jauh, berikut buku yang bisa dibaca atau link yang bisa diakses.

1. George Beekman, Digital Planet: Tomorrow's Technology and You Edisi 10, Penerbit Pearson
2. Penjelasan tentang Sistem Operasi: https://en.wikipedia.org/wiki/Operating_system, https://www.tutorialspoint.com/operating_system/os_overview.htm
3. https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading04.htm

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Wahyono, Budiman Saputra
ISBN : 978-602-244-506-7

Bab 5
Jaringan Komputer dan Internet



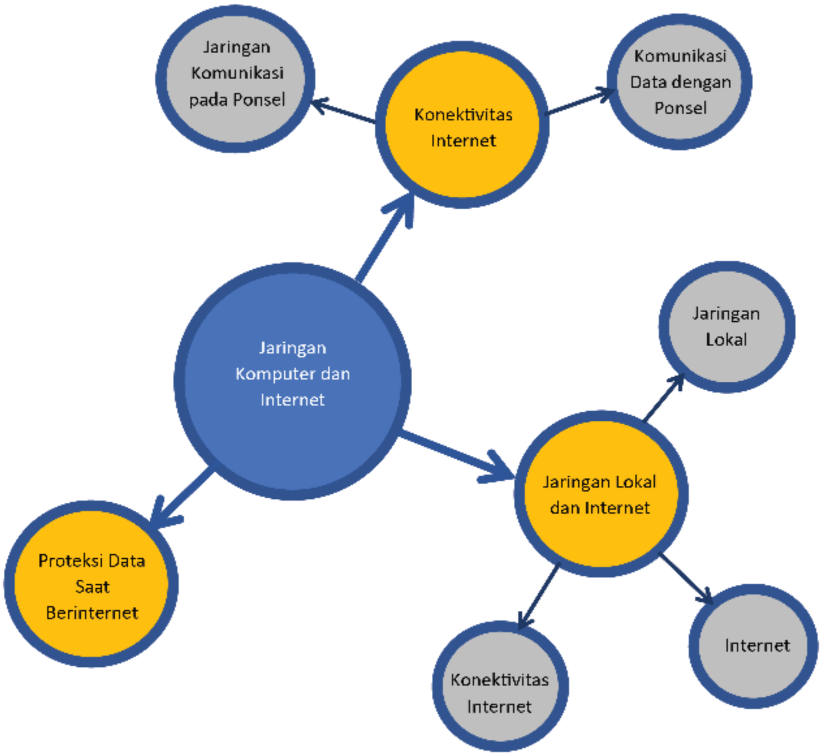
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu memahami (a) perbedaan jaringan lokal dan internet dan jenis-jenis konektivitas internet melalui jaringan kabel dan nirkabel, (b) komunikasi data melalui ponsel, (c) pentingnya proteksi data pribadi saat terhubung ke jaringan internet serta menerapkan enkripsi untuk memproteksi dokumen.

Pertanyaan Pemantik

Tahukah kalian, apa yang akan terjadi jika saat ini tidak ada jaringan internet? Apakah saat kita terhubung ke internet itu selalu aman?

Peta Konsep



Gambar 5.1 Peta Konsep Jaringan Komputer dan Internet

Apersepsi

Internet saat ini merupakan salah satu hal penting yang membantu banyak aspek kehidupan. Saat ini, ketika kita akan membuka rekening baru di bank, kita tidak perlu lagi mengunjungi kantor bank, tetapi cukup mengakses aplikasi bank yang tersedia secara online, mengisi formulir dan melakukan verifikasi secara langsung melalui *video call* dengan petugas. Namun, mengakses internet bukan tanpa masalah. Kita perlu paham cara mengakses internet dengan aman supaya tidak merugikan kita.

Kata Kunci

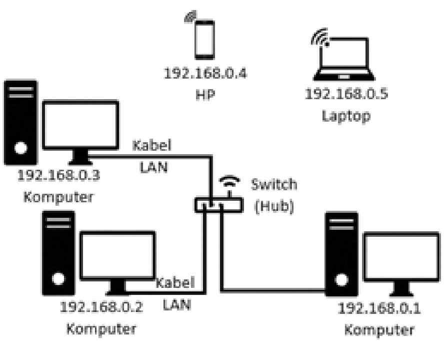
Jaringan lokal, jaringan internet, *internet service provider*, komunikasi data pada HP, Sertifikat SSL.

A. Jaringan Lokal dan Internet

Jaringan komputer menggunakan dua buah atau lebih perangkat dengan menggunakan sebuah sistem komunikasi yang terstandarisasi secara global, yaitu *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP). Jaringan komputer yang menghubungkan komputer ada dua jenis, yaitu jaringan lokal dan jaringan internet. Perbedaan keduanya ada pada jangkauannya. Jaringan lokal memiliki jangkauan yang lebih terbatas dibandingkan dengan jaringan internet.

1. Jaringan Lokal

Jaringan lokal adalah jaringan komputer berkabel maupun nirkabel yang menghubungkan komputer dengan perangkat lainnya dalam area terbatas seperti tempat tinggal, sekolah, laboratorium, kampus universitas, atau gedung kantor. Perangkat atau komputer yang ada di jaringan lokal hanya bisa diakses oleh perangkat lain yang berada pada jaringan yang sama. Setiap perangkat atau komputer yang terhubung dalam jaringan lokal akan memiliki ID unik yang berbeda satu sama lainnya dan disebut Alamat *IP (IP address),* misalnya 192.168.0.1. Dalam implementasinya, biasanya satu perangkat akan disebut *server*, sedangkan perangkat lainnya disebut *client.*



Gambar 5.2 Contoh Jaringan Lokal
Sumber: Dokumen Kemendikbud, 2021

Gambar 5.2 menunjukkan contoh jaringan lokal yang menghubungkan lima buah perangkat, yaitu tiga buah komputer, satu buah HP, dan satu buah laptop. Jika kalian perhatikan, setiap perangkat memiliki *IP address* yang berbeda. Karena *IP address* ini hanya berlaku pada jaringan lokal, bisa juga disebut sebagai *IP private*. Apa yang akan terjadi jika ada dua buah perangkat yang memiliki *IP address* yang sama?

Apa sebenarnya arti dan fungsi dari *IP address*? *IP address* merupakan identitas sebuah komputer dalam jaringan komputer. *IP address* berfungsi sebagai alamat pengiriman data dari satu perangkat ke perangkat lain. Mungkin kalian bisa menganalogikan *IP address* seperti alamat rumah. Saat akan mengirim paket, tentunya akan lebih mudah kalau sudah mengetahui alamat pastinya. Sama halnya dengan ketika kalian akan mengirim data dari satu perangkat ke perangkat lain. Kalian harus tahu data itu akan dikirim ke komputer dengan *IP address* yang mana. *IP address* sendiri biasanya terdiri atas

32 bit yang dipisah menjadi 4 bagian sehingga setiap bagian akan terdiri atas 8 bit. Nah, 8 bit ini merupakan bilangan biner yang diterjemahkan ke dalam bilangan desimal. Tentu kalian sudah belajar tentang bilangan biner. Berikut ini contoh dari *IP address*.

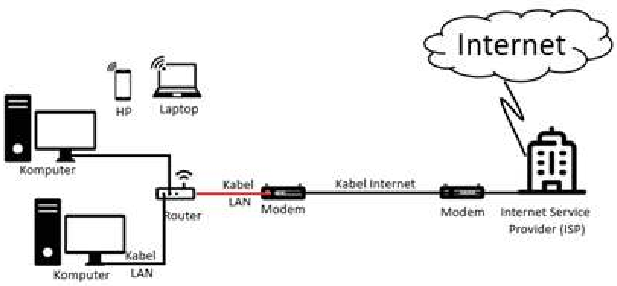
***IP address* dalam bilangan biner**	***IP address***
10101100.11011001.00001010.00001110	**172.217.10.14**

2. Internet

Berbeda dengan jaringan lokal yang hanya menghubungkan perangkat dengan jangkauan yang terbatas pada area tertentu saja misalnya perumahan, perkantoran, sekolah, dan lain sebagainya, jaringan internet memiliki koneksi dengan cakupan yang lebih luas yang bisa menghubungkan perangkat di seluruh dunia. Misalnya, kalian berada di Jakarta. Dengan jaringan internet, kalian bisa mengakses perangkat atau informasi yang ada di Kalimantan bahkan di negara lain. Luar biasa, 'kan?

Internet sendiri merupakan kepanjangan dari *interconnection-networking*, yaitu sebuah jaringan komputer yang menghubungkan banyak perangkat di seluruh dunia. Jaringan internet ini memungkinkan adanya pertukaran data paket (*packet switching communication protocol*) untuk melayani miliaran pengguna di seluruh dunia. Secara konsep, jaringan internet juga disebut jaringan area luas (*Wide Area Network*).

Pada awalnya di tahun 1960-an, internet hanyalah sebagai proyek penelitian yang didanai oleh Departemen Pertahanan Amerika Serikat pada tahun 1969, melalui proyek lembaga ARPA yang mengembangkan jaringan yang dinamakan ARPANET (*Advanced Research Project Agency Network*). Kemudian, berkembang menjadi infrastruktur publik pada 1980-an dengan dukungan dari banyak universitas negeri dan perusahaan swasta.



Gambar 5.3 Jaringan Lokal Bisa Mengubah Menjadi Jaringan Internet
sumber: Dokumen Kemendikbud, 2021

Untuk bisa mengakses jaringan internet, biasanya perangkat apa pun harus terhubung dengan salah satu Internet Service Provider (ISP) dengan menggunakan mekanisme koneksi internet tertentu. Jika jaringan lokal terhubung dengan ISP, baik secara berkabel maupun nirkabel, perangkat dalam jaringan lokal bisa mengakses internet. Hal ini ditunjukkan dengan garis merah (yang diasumsikan sebagai kabel) pada Gambar 5.3. Sama seperti jaringan lokal, setiap perangkat pada jaringan internet juga harus memiliki *IP address* yang berbeda-beda. Karena *IP address* ini berlaku secara global, bisa juga disebut IP public. Akan tetapi, tidak semua perangkat lokal yang terhubung dengan internet memiliki *IP public* karena biasanya *IP public* hanya digunakan pada perangkat utama yang terhubung dengan jaringan internet secara langsung, dalam hal ini ada perangkat modem yang terhubung dengan ISP.

3. Konektivitas Internet

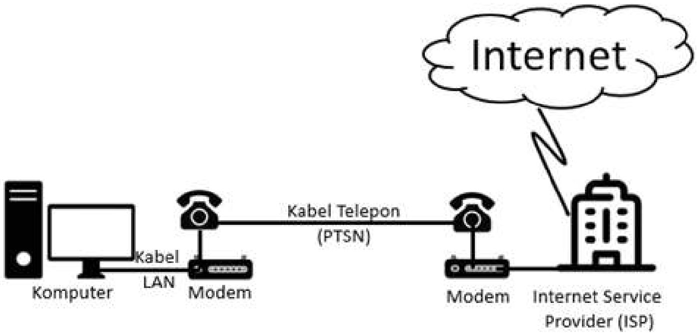
Koneksi dengan internet pada umumnya dilakukan menggunakan kabel internet yang bisa berupa kabel coaxial, kabel fiber optik maupun kabel twisted pair. Meski demikian, kalian juga bisa terhubung ke jaringan internet tanpa menggunakan kabel (*wireless connection*). Baik menggunakan kabel maupun tanpa kabel, keduanya memiliki metode tersendiri cara terkoneksi ke jaringan internet. Berikut ini beberapa jenis koneksi internet yang memungkinkan perangkat kalian terhubung ke dalamnya.

a. Konektivitas Internet pada Jaringan Berkabel

Konektivitas internet pada jaringan berkabel pada umumnya memanfaatkan jaringan kabel telepon maupun televisi yang terhubung dengan Internet Service Provider. Berikut ini beberapa konektivitas pada jaringan berkabel.

a. *Dial-Up PSTN (Public Switched Telephone Network)*

Di awal internet masuk ke Indonesia, teknologi Dial-Up untuk terkoneksi ke internet merupakan metode yang umum digunakan. Untuk terkoneksi dengan internet, Dial-Up memanfaatkan jaringan telepon rumah yang berbentuk kabel. Tiga perangkat yang harus dimiliki untuk koneksi ini ialah komputer, modem, dan sambungan telepon rumah. *Dial-up* ini akan bekerja melalui jalur PSTN (*Public Switched Telephone Network*) hingga bisa terhubung dengan ISP (*Internet Service Provider*). Kecepatan akses jenis koneksi internet *Dial-Up* hanya 12 hingga 20 Kbps, maksimum hanya 56 Kbps (*kilobyte per second*). Gambar 5.4 menunjukan ilustrasi bagaimana konektivitas internet menggunakan cara *Dial-Up.*



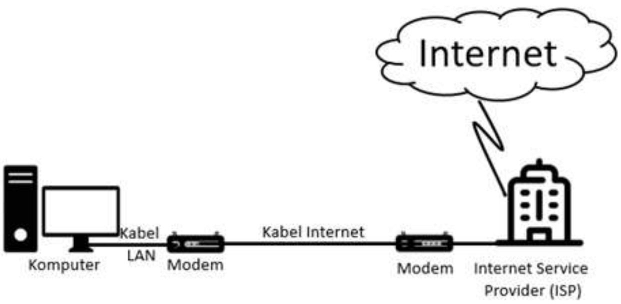
Gambar 5.4 Diagram Konektivitas Internet Jaringan Berkabel melalui Dial-Up dan ASDL
sumber: Dokumen Kemendikbud, 2021

b. *ADSL (Asymmetric Digital Subscriber Line)*

Sama seperti pada *Dial-Up*, koneksi internet ADSL juga bekerja dengan menggunakan teknologi modem. Modem yang digunakan bekerja pada frekuensi antara 34 kHz sampai 1104 kHz, berbeda dengan modem konvensional yang bekerja pada frekuensi di bawah 4 kHz. Dengan ADSL, kita bisa mengirimkan suara hingga fax ke pengguna lainnya dengan menggunakan rentang frekuensi yang berbeda. Konektivitas internet dengan ADSL pada prinsipnya sama dengan *Dial-Up* yang menggunakan jaringan telepon kabel seperti ditunjukan pada Gambar 5.4, tetapi yang membedakan ialah jenis modemnya.

b. LAN (*Local Area Network*)

LAN termasuk koneksi internet yang banyak dikenal saat ini. Sistem kerjanya ialah menggunakan satu komputer sebagai *server* yang terhubung dengan internet menggunakan kabel telepon atau antena melalui *Internet Service Provider.* Kemudian, komputer lainnya hanya perlu terkoneksi dengan *server* untuk bisa mengakses internet dengan memakai kartu LAN (*LAN Card*) dan kabel koaksial (UTP). Jaringan koneksi internet ini hanya mencakup wilayah yang sangat kecil. Ilustrasi jaringan internet berkabel LAN ditunjukkan pada Gambar 5.5.



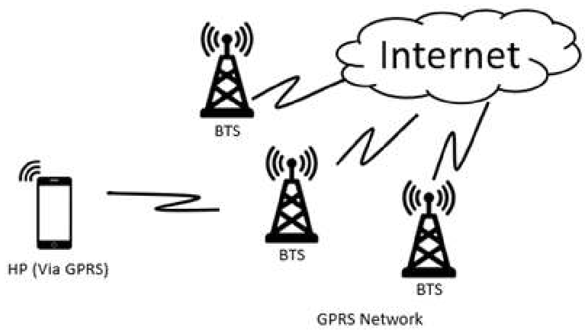
Gambar 5.5 Konektivitas Jaringan LAN
Sumber: Dokumen Kemendikbud, 2021

b. Konektivitas Internet pada Jaringan Nirkabel

Konektivitas internet pada jaringan nirkabel merupakan jaringan internet yang tidak menggunakan kabel untuk menghubungkan satu perangkat dan perangkat lain. Jaringan nirkabel ini sering dipakai untuk jaringan komputer baik pada jarak yang dekat (beberapa meter, memakai alat/ pemancar *bluetooth*) maupun pada jarak jauh (lewat satelit). Jaringan nirkabel biasanya menghubungkan satu sistem komputer dan sistem yang lain dengan menggunakan beberapa macam media transmisi tanpa kabel, seperti: gelombang radio, gelombang mikro, maupun cahaya inframerah. Berikut ini beberapa konektivitas pada jaringan nirkabel.

a. GPRS *(General Packet Radio Service)*

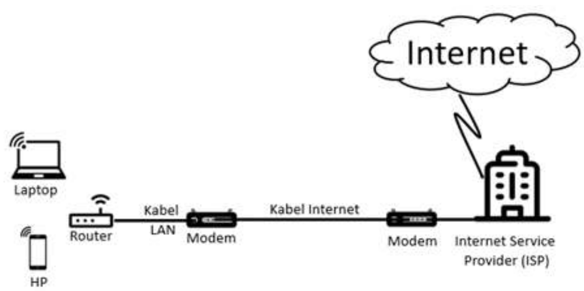
Jenis koneksi internet GPRS (*General Packet Radio Service*) menggunakan gelombang radio untuk komunikasi data dan suara. GPRS ini mempunyai kemampuan untuk mengirimkan data dan suara pada alat komunikasi bergerak. Sistem GPRS ini bisa dipakai untuk transfer data, berbentuk paket data yang terkait dengan pengiriman surel (surat elektronik) hingga berselancar di dunia maya. Layanan GPRS ini dipasang pada jenis ponsel dengan tipe GSM (*Global Systems for Mobile Communications*). Ilustrasi jaringan internet nirkabel GPRS ditunjukkan pada Gambar 5.6.



Gambar 5.6 Ilustrasi Konektivitas Jaringan Nirkabel via GPRS
Sumber: Dokumen Kemendikbud,2021

b. Wifi

Wifi merupakan singkatan dari *wireless fidelity* yang merupakan sistem standar yang digunakan untuk terkoneksi dengan internet tanpa menggunakan kabel. Teknologi ini memakai frekuensi tinggi, berada pada spektrum 2,4 GHz.

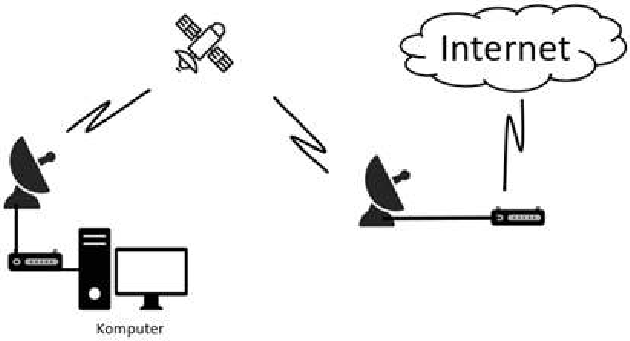


Gambar 5.7 Ilustrasi Konektivitas Jaringan Nirkabel Wifi
Sumber: Dokumen Kemendikbud, 2021

Salah satu keunggulan jenis koneksi internet *wifi* ialah praktis karena tidak perlu memasang kabel jaringan. Namun, koneksi *wifi* memiliki jangkauan terbatas. Ilustrasi jaringan internet nirkabel *wifi* ditunjukan pada Gambar 5.7.

c. Akses Satelit

Jenis koneksi internet ini merupakan layanan internet yang memakai antena parabola sehingga kecepatan akses cukup tinggi. Namun, biasanya, jenis koneksi internet ini memang mahal terutama di Indonesia, meski sepadan dengan kecepatan aksesnya. Ilustrasi jaringan internet nirkabel satelit ditunjukkan pada Gambar 5.8.



Gambar 5.8 Ilustrasi Konektivitas Jaringan Nirkabel melalui Satelit
sumber: Dokumen Kemendikbud,2021



Aktivitas Individu

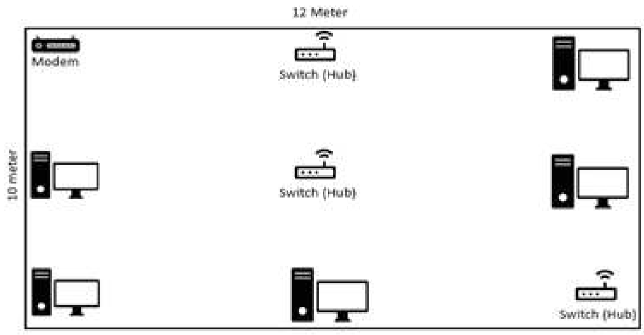
Aktivitas JKI-K10-01-U Perancangan Jaringan Komputer

Pada aktivitas ini, kalian akan mencoba merancang konfigurasi sebuah jaringan komputer yang menghubungkan perangkat komputer sehingga bisa berkomunikasi dengan perangkat komputer lainnya.

Deskripsi Kasus

Ani berencana memasang jaringan internet baru pada suatu *Internet Service Provider*. ISP tersebut menyediakan satu buah modem secara gratis yang memiliki *IP public* yang hanya bisa terkoneksi dengan maksimal dua perangkat lainnya. Adapun Ani sendiri memiliki tiga buah *switch/router* yang masing-masing bisa terpasang dengan maksimal tiga perangkat lainnya. Ani menginginkan agar semua perangkat komputer yang ada di rumahnya terkoneksi dengan internet. Namun, dia juga menginginkan agar biaya pasang yang dikeluarkan semurah mungkin. Biaya pasang ini bergantung pada total panjang kabel yang harus digunakan. Jika diberikan posisi semua perangkat

yang dimiliki oleh Ani seperti pada Gambar 5.9 di bawah, bantulah dia untuk memodelkan konfigurasi jaringan di rumahnya sehingga biaya yang harus dikeluarkan semurah mungkin. Posisi perangkat tidak boleh dipindah-pindah dan semua *switch/router* tidak harus digunakan.



Gambar 5.9 Ilustrasi Studi Kasus Perancangan Jaringan Komputer
Sumber: Dokumen Kemendikbud, 2021

Apa yang kalian perlukan?

1. Kertas kosong untuk mengonfigurasi jaringan
2. Spidol

Apa yang harus kalian lakukan?

1. Buatlah model jaringan komputer pada gambar di atas.
2. Setelah memodelkan konfigurasi jaringan, bandingkan dengan rancangan teman kalian. Apakah sama? Mana yang lebih baik?
3. Selanjutnya, buatlah sketsa denah rumah kalian. Bayangkan kalian memiliki beberapa komputer yang harus terkoneksi dengan internet. Buatlah model konfigurasi jaringannya.
4. Setelah selesai membuat model konfigurasi jaringan komputer, jangan lupa kalian harus meminta kepada teknisi jaringan jika ingin memasangnya. Pasti teknisi akan sangat senang karena kalian telah membuatkan modelnya sehingga teknisi hanya perlu memasangnya.

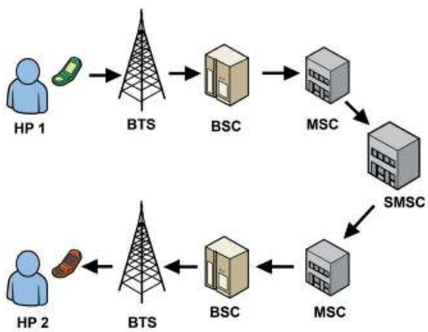
B. Komunikasi Data dengan Ponsel

Salah satu penggunaan jaringan internet dalam membantu kehidupan sehari-hari ialah sebagai sarana komunikasi baik melalui ponsel maupun melalui perangkat lainnya. Oleh sebab itu, kalian perlu memahami tentang mekanisme komunikasi data yang terjadi pada ponsel. Di lain pihak, ketika menggunakan internet, kalian juga perlu memperhatikan keamanan data yang dikirim melalui jaringan internet.

1. Jaringan Komunikasi pada Ponsel

Komunikasi data merupakan sebuah mekanisme pengiriman dan penerimaan data dari satu perangkat ke perangkat lain. Ketika kita menggunakan ponsel, baik untuk keperluan kirim pesan atau SMS (*short message system*), panggilan suara, maupun transfer dokumen, di dalamnya ada mekanisme komunikasi data yang terjadi antara ponsel pengirim dan ponsel penerima. Komunikasi data dalam ponsel dilakukan pada BTS (*Base Transceiver Station*) melalui beberapa cara dan media, yaitu dengan *bluetooth*, kabel data ataupun melalui jaringan komunikasi seperti CDMA (*Code Division Multiple Access*) atau GSM (*Global System for Mobile*). Namun, karena makin berkembangnya teknologi internet, saat ini, komunikasi data menggunakan ponsel bisa melalui jaringan internet antara lain seperti berikut.

a. GPRS (*General Packet Radio Service*) merupakan jaringan internet pada ponsel yang memiliki kecepatan antara 35-171 Kpbs. Dengan kecepatan ini, biasanya, hanya cukup untuk mengirim pesan yang tidak terlalu panjang.
b. EDGE (*Enhanced Data rate for GSM Evolution*) merupakan jaringan internet yang memiliki kecepatan lebih baik dibandingkan dengan GPRS karena memiliki kecepatan antara 120-384 Kbps.
C. 3G merupakan jaringan internet GPRS versi 3 yang menggunakan protokol transfer data *United Mobile Telecommunication Technology* sehingga menghasilkan kecepatan antara 384 Kbps – 2 Mbps.
d. Generasi berikutnya ialah HSPA (*High Speed Packet Access*) yang dikenal dengan nama 3.5G dan memiliki kecepatan 600 Kbps – 10 Mbps. Adapun HSDPA (*High Speed Downlink Packet Access*) atau dikenal dengan nama H+ pada layar HP kita memiliki kecepatan yang stabil pada kisaran 7.2 Mbps.
e. 4G/LTE (*Long Term Evolution*) merupakan generasi keempat dari GPRS yang memiliki kecepatan hingga 100 Mbps. Saat ini, hampir sebagian besar ponsel di pasaran sudah mendukung jaringan 4G/LTE.



Gambar 5.10 Salah Satu Alur Kerja Proses Komunikasi Data SMS melalui Ponsel
Sumber: Dokumen Kemendikbud, 2021

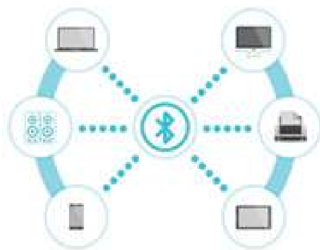
Gambar 5.10 menunjukkan ilustrasi cara kerja komunikasi data melalui ponsel, misalnya pada proses ketika kita mengirimkan SMS (*Short Message System*). Ketika pengguna mengirim SMS, pesan dikirim ke MSC melalui jaringan seluler yang tersedia yang meliputi tower BTS yang sedang meng-handle komunikasi pengguna, lalu ke BSC (*Base Station Controller*), kemudian sampai ke MSC (*Mobile Switching Center*). MSC kemudian meneruskan lagi SMS ke SMSC (*Short Message Service Center*) untuk disimpan. SMSC kemudian mengecek (lewat HLR – *Home Location Register*) untuk mengetahui apakah ponsel tujuan sedang aktif dan di manakah ponsel tujuan tersebut.

Jika ponsel sedang tidak aktif, pesan tetap disimpan di SMSC itu sendiri, menunggu MSC memberitahukan bahwa ponsel sudah aktif kembali untuk kemudian SMS dikirim dengan batas maksimum waktu tunggu, yaitu *validity period* dari pesan SMS itu sendiri. Jika ponsel tujuan aktif, pesan disampaikan MSC lewat jaringan yang sedang menangani penerima (BSC dan BTS).

Sebenarnya, di dalam kebanyakan ponsel dan model GSM/CDMA, terdapat suatu komponen *wireless modem/engine* yang dapat diperintah antara lain untuk mengirim suatu pesan SMS dengan protokol tertentu. Standar perintah tersebut dikenal sebagai AT-Command, sedangkan protokolnya disebut PDU (*Protocol Data Unit*). Melalui AT-Command dan PDU inilah, kita dapat membuat komputer/mikrokontroler mengirim/menerima SMS secara otomatis berdasarkan program yang kita buat.

2. Komunikasi Data dengan Ponsel

Ponsel/*handphone*/HP merupakan sebuah perangkat yang bekerja layaknya sebuah komputer karena bisa terhubung ke dalam sebuah jaringan lokal maupun jaringan internet. Oleh karena itu, sangat memungkinkan bagi kalian untuk saling melakukan komunikasi data antara satu ponsel dan ponsel lainnya. Ada beberapa jenis koneksi jaringan pada perangkat ponsel yang digunakan untuk komunikasi data. Sebagian besar koneksi tersebut lebih banyak menggunakan media tanpa kabel, yaitu antara lain koneksi dengan kabel data, koneksi dengan *bluetooth*, dan koneksi dengan *wifi*.

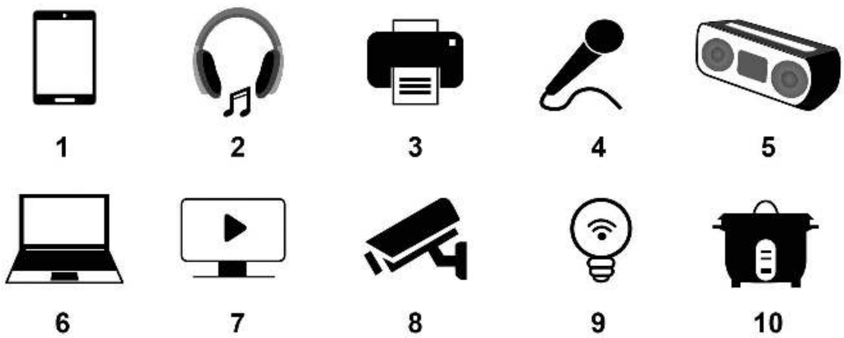
Koneksi dengan Kabel	Koneksi dengan kabel pada ponsel biasanya dilakukan dengan menggunakan *data cable* (kabel data). Penggunaannya lebih banyak untuk menghubungkan ponsel dan perangkat komputer atau perangkat lainnya yang memiliki *port* USB. Karena menggunakan kabel, jangkauan koneksi ini sangat terbatas sesuai dengan panjang kabel data yang kalian miliki.	 Gambar 5.11. Kabel Data
Koneksi dengan *Bluetooth*	*Bluetooth* adalah komunikasi data tanpa kabel yang menggunakan gelombang elektromagnetik yang beroperasi pada frekuensi 2.45. *Bluetooth* memungkinkan pengguna perangkat seperti ponsel, *printer*, *mouse*, *keyboard* dan perangkat lainnya bisa terkoneksi tanpa kabel dengan jangkauan sekitar 1 meter.	 Gambar 5.12. Contoh Ilustrasi Komunikasi Data Menggunakan Bluetooth
Koneksi dengan *wifi*	Koneksi data dengan jaringan tanpa kabel menggunakan teknologi *wifi* merupakan jenis koneksi yang banyak digemari. Alasan utamanya ialah karena *bandwidth*-nya yang besar dan hemat energi baterai. Kapasitas *bandwidth* maksimum sangat bergantung pada jarak dan *data rate* dari perangkat *access point*. Kekurangannya ialah jangkauannya yang tidak jauh, maksimum 100 meter, tetapi dengan *bandwidth* yang minimum (*data rate* 1Mbps).	 Gambar 5.13 Contoh Ilustrasi Komunikasi Data Menggunakan Wifi Sumber: Dokumen Kemendikbud, 2021

Ayo Identifikasi

Aktivitas Kelompok

Aktivitas JKI-K10-02 Identifikasi Jenis Koneksi Data via Ponsel

Pada aktivitas ini, kalian diminta untuk mengidentifikasi jenis koneksi yang bisa dilakukan antara ponsel dan perangkat lain. Berilah tanda centang pada jenis koneksi data jika bisa digunakan untuk komunikasi data antara ponsel dengan perangkat di bawah ini.



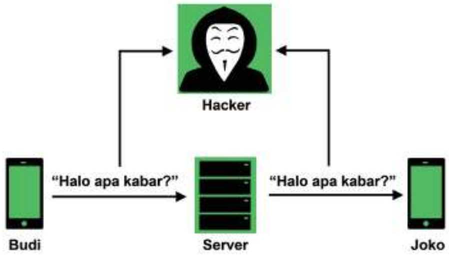
Gambar 5.14 Ilustrasi Berbagai Piranti Elektronik

Perhatikan gambar-gambar perangkat di atas! Pernahkah kalian melihat atau menggunakan perangkat tersebut? Menurut kalian, apakah perangkat-perangkat tersebut bisa terhubung dengan ponsel yang kalian miliki? Bergantung pada spesifikasinya, setiap perangkat bisa saja terhubung dengan ponsel menggunakan salah satu koneksi Kabel Data, Bluetooth, atau Wifi, atau gabungan dari ketiganya. Identifikasi gambar-gambar di atas sesuai dengan perangkat yang kalian miliki atau pengamatan kalian di lingkungan sekitar misalnya sekolah atau rumah. Manakah perangkat yang bisa terhubung dengan ponsel dan mana yang tidak bisa terhubung dengan ponsel? Jangan lupa untuk menulis merk atau tipe perangkat yang kalian amati. Jika tidak punya salah satu perangkat, kalian bisa saling berdiskusi dengan teman kalian yang mungkin memiliki perangkat tersebut.

No	Nama Perangkat	Merk/Tipe	Cara Koneksi dengan Ponsel
1			
2			
3			
4			
5			
...			

C. Proteksi Data Saat Berinternet

Apakah kalian menggunakan media sosial sebagai salah satu sarana komunikasi? Tahukah kalian bahwa saat menggunakan media sosial, pesan yang kalian kirimkan itu dienkripsi? Ini artinya, setiap pesan atau panggilan yang dilakukan melalui media sosial, dienkripsi untuk melindungi pesan supaya tidak diketahui oleh orang lain. Gambar 5.15 menunjukkan ilustrasi bahayanya jika pesan tidak dienkripsi.



Gambar 5.15 Ilustrasi Jika Pesan di Media Sosial Tidak Dienkripsi
Sumber: Dokumen Kemendikbud, 2021

Seperti terlihat pada Gambar 5.15, pesan yang dikirim oleh Budi berpotensi diketahui oleh orang lain sehingga kerahasiaan data jadi tidak terproteksi dengan baik. Inilah pentingnya enkripsi data ketika kalian terkoneksi dengan jaringan internet. Kalian harus memastikan bahwa aplikasi messenger yang kalian gunakan memiliki fitur enkripsi supaya data yang kalian kirimkan aman terproteksi dan tidak bisa diketahui orang lain.

Ketika kalian membuka sebuah situs web, kemudian melakukan log in ke dalam situs web tersebut dengan memasukan nama pengguna dan kata sandi, apakah kalian tahu bahwa data yang kalian masukkan aman dan tidak mungkin bisa diketahui orang lain? Untuk memastikan ini, kalian harus paham bahwa situs web yang kalian kunjungi itu aman atau tidak. Cara paling mudah untuk mengetahui situs web yang kalian kunjungi itu aman karena melakukan proteksi pada data yang kalian input ialah dengan melihat di alamatnya apakah didahului dengan HTTPS dan ada icon seperti kunci di sebelah alamat situs web, seperti ditunjukkan pada Gambar 5.15. Apakah kalian tahu bedanya http dan https? *Hypertext Transfer Protocol* (HTTP) adalah protokol yang mengatur komunikasi antara *client* (komputer pengguna) dan *server* (komputer yang menyimpan data yang diakses melalui situs web). Pada umumnya, cara komunikasi antara *client* dan *server* ialah *client* melakukan request ke *server*, kemudian *server* mengirimkan respons terhadap situs web. Respons yang dimaksud dapat berupa *file* yang akan ditampilkan di *browser* yang kalian gunakan. Semua kegiatan tersebut diatur oleh suatu

protokol HTTP. Sayangnya, HTTP tidak menjamin keamanan data sehingga ada potensi data bisa dimanipulasi. Adapun HTTPS adalah versi HTTP yang lebih aman yang mampu menjaga keamanan data yang sedang diproses.



Gambar 5.16 Situs kemdikbud.go.id Menggunakan Sertifikat SSL
Sumber: Dokumen Kemendikbud, 2021

Teknologi enkripsi berbasis https dinamakan SSL yang merupakan singkatan dari *Secure Sockets Layer*. SSL adalah suatu teknologi keamanan standar global yang memungkinkan komunikasi terenkripsi antara peramban web dan server web. SSL banyak digunakan pada berbagai situs web untuk mengurangi risiko informasi yang bersifat sensitif (misalnya, nomor kartu kredit, nama pengguna, kata sandi, email, dll) dari pencurian atau perusakan oleh peretas dan pencuri identitas. Sertifikat SSL berfungsi untuk mengotentikasi identitas situs web untuk menjamin pengunjung bahwa situs web tersebut bukan merupakan situs web palsu sekaligus mengenkripsi data yang sedang dikirim. Pada *website* yang dilengkapi dengan SSL, ketika kalian menekan ikon kunci, akan muncul informasi tentang validitas dari sertifikat tersebut seperti ditunjukkan pada Gambar 5.16. Jadi, sebelum kalian mengunjungi sebuah situs web, pastikan bahwa situs web tersebut aman.

Uji Kompetensi

Pilihan Ganda

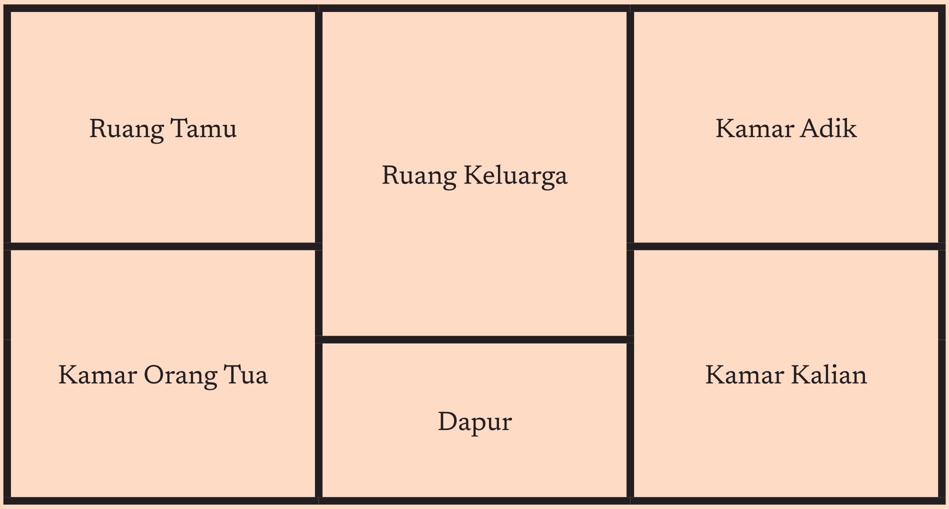
1. Diberikan konfigurasi jaringan sebagai berikut.
   Jika diasumsikan bahwa ketika melakukan transfer data pada jaringan internet, makin panjang kabel yang harus dilalui biasanya membutuhkan waktu yang makin lama, ini bisa saja berpotensi pada lambatnya kecepatan internet. Dari gambar jaringan di atas, manakah komputer yang akan memiliki kecepatan internet paling lambat?
   A. Hanya Komputer B
   B. Komputer B dan C
   C. Komputer C dan E
   D. Hanya Komputer C

2. Sesuai Gambar pada soal nomor 1, manakah komputer yang akan memiliki kecepatan internet paling cepat?
   A. Hanya Komputer A
   B. Komputer B dan D
   C. Komputer C dan E
   D. Hanya Komputer B
3. Konektivitas internet berkabel yang memanfaatkan jaringan kabel telepon untuk terhubung dengan jaringan internet antara lain ialah ...
   A. Dial-Up dan GPRS
   B. ADSL dan Dial-Up
   C. GRPS dan TV Kabel
   D. Satelit dan Wifi
4. Diketahui bahwa Budi dan Chika berada di suatu lokasi dengan jarak 1 meter, tetapi dipisahkan oleh tembok. Karena suatu hal, Budi ingin mengirim *file* kepada Cika melalui HP mereka, yang sama-sama memiliki fasilitas koneksi kabel data, *bluetooth* dan *wifi*. Menurut kalian koneksi apa yang bisa mereka gunakan untuk saling mengirimkan *file* pada kasus ini?
   A. Hanya koneksi *wifi*.
   B. Hanya koneksi *bluetooth*.
   C. Koneksi *wifi* dan *bluetooth*.
   D. Koneksi *wifi* dan kabel data.
5. Sertifikat SSL merupakan salah satu cara yang bisa digunakan untuk memproteksi data pada jaringan internet. Berikut ini yang merupakan fungsi lain dari Sertifikat SSL adalah . . . .
   A. melakukan autentikasi pada sebuah situs web untuk mengecek apakah situs web tersebut bisa diakses dengan cepat
   B. memastikan sebuah situs web yang dikunjungi merupakan situs web asli yang terjamin keamananya
   C. mencegah adanya *hacker* yang mencuri data pribadi
   D. semua jawaban benar

Uraian

Misalnya kalian akan membuat sebuah jaringan komputer di lingkungan rumah kalian supaya komputer yang berada di kamar kalian, kamar orang tua, dan kamar adik bisa terhubung ke internet. Jika diketahui posisi

modem ISP ada di ruang tamu dan diberikan denah kamar yang ada di rumah kalian sebagai berikut, gambarkan konfigurasi jaringan yang tepat sehingga menggunakan total panjang kabel paling pendek supaya terlihat rapi!



Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Apakah kalian telah telah memahami konsep jaringan lokal dan jaringan internet?
2. Apakah kalian telah memahami konektivitas internet berkabel dan nirkabel?
3. Bayangkan jika tidak ada jaringan komputer. Bayangkan jika semua komputer hanya sendiri-sendiri (*stand alone*) dan tidak terhubung satu sama lain. Apa yang akan terjadi?
4. Akibat adanya konektivitas perangkat, apa saja yang harus diwaspadai terkait data yang kita kirimkan dari perangkat kita?

Ingin Tahu Lebih?

Jika kalian tertarik dengan materi ini dan ingin mendalaminya lebih jauh, berikut buku yang bisa dibaca atau link yang bisa diakses.

1. George Beekman, Digital Planet: Tomorrow's Technology and You, Prentice Hall, 2012
2. https://id.wikipedia.org/wiki/Jaringan_komputer
3. https://id.wikipedia.org/wiki/Internet_Sehat_dan_Aman
4. How does Internet work, https://www.youtube.com/watch?v=7_LPdttKXPc
5. https://id.wikipedia.org/wiki/Transport_Layer_Security
6. https://id.wikipedia.org/wiki/HTTPS

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Hanson Prihantoro Putro
ISBN : 978-602-244-506-7

Bab 6
Analisis Data



Sumber: https://www.michiganstateuniversityonline.com/resources/business-analytics/types-of-data-analytics-and-how-to-apply-them/

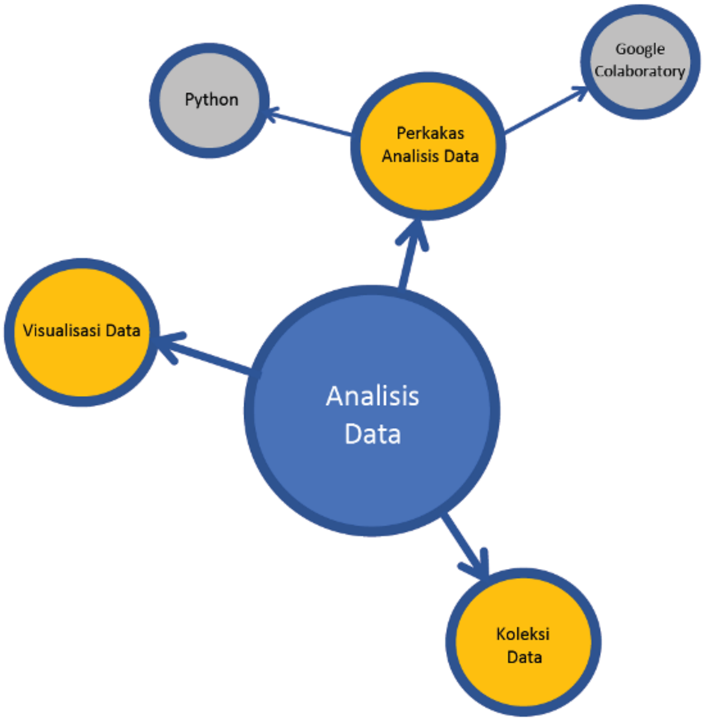
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu (a) menggunakan alat bantu untuk menulis, menjalankan, dan mengembangkan program analisis data; (b) memahami pengkoleksian data melalui berbagai cara, khususnya secara otomatis melalui perangkat; (c) memahami transformasi data; (d) melakukan interpretasi data dan memahami aspek privasi dan keamanan data.

Pertanyaan Pemantik

Bagaimana cara untuk mengumpulkan banyak data secara otomatis dan menampilkannya agar kita mudah memahami data tersebut?

Peta Konsep



Gambar 6.1 Peta Konsep Analisis Data

Apersepsi

Apakah kalian pernah memakai mesin pencari seperti Google, Bing, Yahoo atau lainnya untuk mencari data? Dengan mengetikkan satu atau lebih kata kunci, semua halaman web yang mengandung apa yang kalian ketikkan tersebut muncul. Bagaimana mesin pencari melakukan itu?

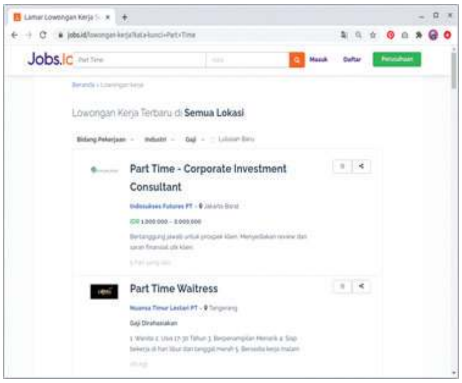
Kata Kunci

Analisis Data, Google Colaboratory, Koleksi Data, Web Scraping, Visualisasi Data

Pengantar Analisis Data

Pernahkah kalian mengalami saat sedang mencari informasi menemukan bahwa informasinya bertebaran di beberapa, bahkan di banyak halaman web? Mungkin pada saat itu kalian melakukan pencarian dengan mesin pencari dan mendapatkan hasil berupa daftar halaman yang memuat informasi yang kalian cari. Contohnya misalnya kalian ingin membanding-bandingkan harga barang dari beberapa situs toko *online*. Kalian harus membuka halamannya satu per satu, kemudian membandingkan, dan memutuskan akan membeli barang sesuai dengan kriteria yang kalian tetapkan. Jika membuka semua halaman web tersebut harus dilakukan manual, akan sangat melelahkan dan membosankan bukan? Bayangkan jika ada robot yang dapat melakukannya untuk kalian, sehingga mudah untuk menganasis. Tentunya pengambilan keputusan akan menjadi lebih cepat. Robot itu tidak harus berwujud seperti manusia *lho*. Robot dapat berupa sebuah perangkat lunak komputer. Nah, pada unit ini kalian akan membuat sebuah robot seperti ini yang akan membantu kalian.

Setelah lulus kuliah, apa yang kalian lakukan jika ingin mencari pekerjaan? Atau mungkin selama kuliah juga ingin mencari pekerjaan sambilan (*part time*)? Untuk mencari lowongan pekerjaan, kita menggunakan koran atau pun halaman web, misalnya JobsID. Coba ketik https://www.jobs.id/ di *browser*. Di halaman ini, kalian dapat mencari pekerjaan dengan memasukkan kata kunci pekerjaan di *form* pencarian yang ada, misalnya pekerjaan sambilan atau "*part time*", seperti diperlihatkan pada Gambar 6.2.



Gambar 6.2 Contoh Tampilan Halaman Website Pencarian Kerja

Sumber: Dokumen Kemendikbud, 2021

Dari halaman tersebut kita peroleh daftar pekerjaan yang ditampilkan cukup detail dalam beberapa halaman. Dengan tampilan tersebut, kita perlu melakukan *scroll* sekitar delapan kali untuk mendapatkan sekitar 23 lowongan pekerjaan. Bagaimana caranya kita dapat merangkum lowongan pekerjaan tersebut sehingga lebih mudah untuk melihat dan memilih pekerjaan yang kita inginkan?

Dalam kajian analisis data, terdapat teknik yang disebut sebagai *scraping*. *Scraping* adalah salah satu bentuk penyalinan, di mana data tertentu dikumpulkan dan disalin dari sebuah halaman web, bisa ke dalam basis data, *spreadsheet* atau tampilan tertentu untuk pengambilan atau analisis data. *Scraping* bisa dilakukan dengan menggunakan sebuah bahasa pemrograman yang mendukung. Dengan

scraping, kita bisa mendapatkan rangkuman dari suatu halaman web sehingga 23 lowongan yang sebelumnya ditampilkan dalam beberapa *scroll* kini menjadi diringkas dalam satu tampilan dengan data penting yang kita inginkan saja. Dalam aktivitas di unit pembelajaran ini, kita akan melakukan scraping dengan membuat program yang mengambil data dari sebuah halaman *website.*

Untuk membangun sebuah *scraper*, kita perlu menentukan bahasa pemrograman yang memudahkan kita. Setiap bahasa pemrograman akan diimplementasi oleh sebuah lingkungan pengembangan terintegrasi (*Integrated Development Environment*/IDE) tertentu. Ada banyak pilihan bahasa pemrograman beserta IDE-nya. Disini kita akan menggunakan bahasa pemrograman Python. Bahasa Python dipilih karena menjadi bahasa yang sering digunakan dalam analisis data. Python memiliki banyak sekali fungsi dan *library (*pustaka) yang memudahkan kita untuk melakukan analisis data, salah satunya untuk melakukan *scraping* ini.

Kemudian untuk membuat program Python, kita memerlukan alat bantu IDE atau lingkungan kerja untuk menulis dan menjalankan program Python tersebut. Salah satu alat bantu yang bisa kita gunakan ini yaitu *Google Collaboratory* atau *Google Colab*. *Google Colab* dipilih karena dapat digunakan secara online untuk mengambil data dari *website* yang *online* pula. Selain itu, banyak fungsi dan pustaka Python yang sudah terpasang dalam *Google Colab* sehingga bisa langsung digunakan.

Kita akan mulai dengan pengenalan alat bantu *Google Colab* dan Python. Setelah cukup mengenal dua alat bantu analisis data ini, mulailah kita membuat proyek untuk *scraping* halaman *website* yang dipilih. Kemudian setelah itu, data hasil *scraping* ini divisualisasikan ke dalam berbagai bentuk diagram dan *chart*.

A. Pengenalan Perkakas Analisis Data

1. Google Colaboratory (Google Colab)

Ayo Mulai

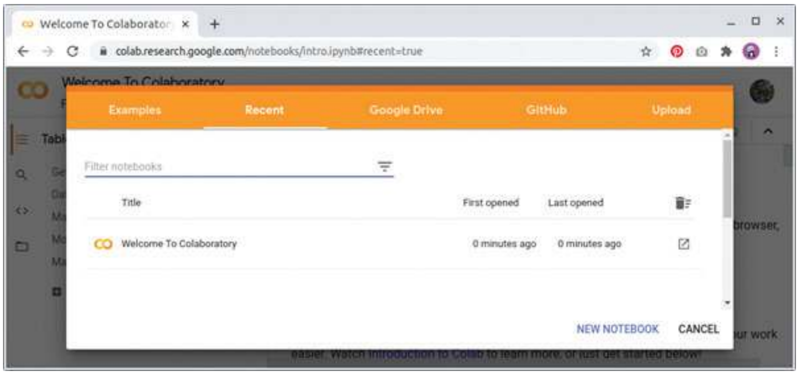
Aktivitas AD-K10-01-P: Mengenal Google Colab

Aktivitas ini dilakukan untuk memulai mengenal alat/lingkungan analisis data, salah satunya ialah Google Colab. Google Colab atau Google Colaboratory adalah salah satu lingkungan pengembangan aplikasi terintegrasi yang disediakan oleh Google secara *online* (*Online* IDE). Karena sifatnya yang *online*, maka pengguna tidak perlu melakukan instalasi dan

dapat langsung menggunakan Google Colab untuk menulis program dan melakukan pengolahan data dari Internet. Selain itu, Google Colab juga memiliki banyak fungsi serta *library* yang dapat digunakan untuk membantu pengolahan data, termasuk untuk melakukan *scraping*.

Sebelum membuat program, berikut beberapa langkah mengenal penggunaan Google Colab.

1. Buka Google Colaboratory melalui link https://colab.research.google.com/ Jika diminta untuk *Sign-In*, silakan masuk dengan menggunakan akun Google/GMail.
2. Buat catatan baru melalui pilihan NEW NOTEBOOK seperti diperlihatkan pada Gambar 6.3. *Notebook* adalah penamaan untuk *file* kerja di dalam Google Colab, tempat membuat berbagai macam dokumen, termasuk teks dokumen/catatan dan teks kode program Python.

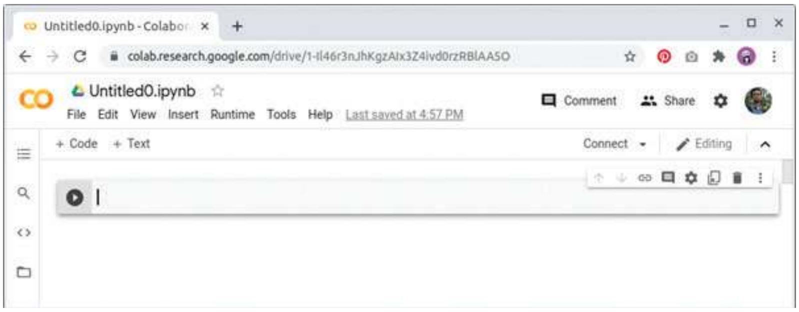


Gambar 6.3 Tampilan Google Colab Saat Dibuka Pertama Kali
Sumber: Dokumen Kemendikbud, 2021

Perhatikan bahwa Google Colab memiliki beberapa bagian area kerja yang hampir serupa dengan berbagai IDE yang lain. Beberapa objek yang ada di Google Colab diperlihatkan pada Gambar 6.3, yaitu seperti berikut.

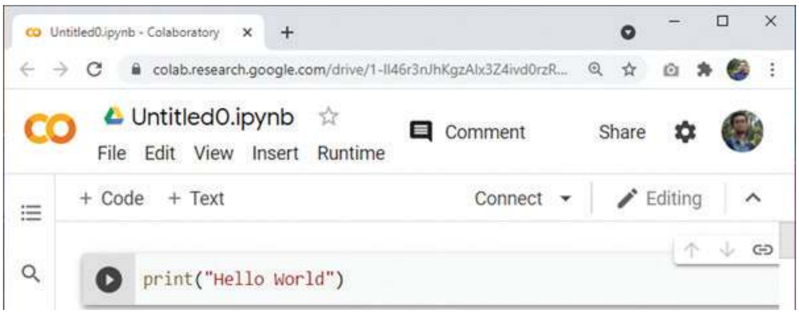
a. Nama *Notebook*, yang merupakan nama *file* berekstensi ipynb (Ipython Notebook). Kita bisa mengganti nama *notebook* dengan mengklik nama notebook tersebut.
b. *Star*, yang menandakan tingkat kepentingan file ini.
c. *Header* Kanan, yang berisi pilihan komentar, pilihan untuk berbagi (*share*), pengaturan (*setting*) file, serta pengaturan akun.
d. Menubar, yang berisi pilihan menu seperti *File, Edit, View* dan lain sebagainya.
e. Panel Kiri, yang berisi beberapa ikon menu seperti Daftar Isi, Pencarian, Penyisipan Kode, Pengelolaan File, dan Pemilihan Perintah.
f. *Toolbar* Atas, yang berisi pilihan ikon untuk Penambahan Kode atau Teks, Pilihan Koneksi, dan Pilihan Menutup Menubar.

g. Konten *Notebook*, yang berisi tulisan kode program atau teks yang kita tulis.
h. *Cell Toolbar,* yang berisi pilihan ikon untuk mengatur sel pada konten Pemindahan Atas atau bawah, Koneksi Antarsel, Penambahan komentar, Pengaturan Editor, Penggandaan, Penghapusan Sel dan lain sebagainya.



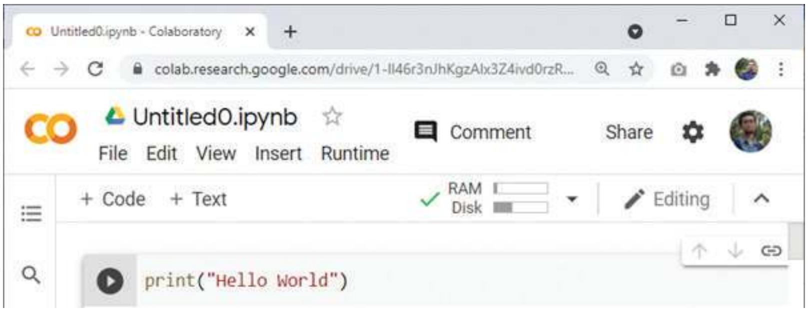
Gambar 6.4 Area Kerja Google Colab
Sumber: Dokumen Kemendikbud, 2021

3. Mulailah untuk membuat kode program Python dengan mencetak sebuah teks seperti contoh pada Gambar 6.5. Ketik satu baris kode tersebut di area Konten Notebook.



Gambar 6.5 Contoh sederhana kode program Python
Sumber: Dokumen Kemendikbud, 2021

4. Untuk menguji hasil dari kode program tersebut, tekan tombol "Run cell" (bulat dan segitiga) yang ada di pojok kiri area Konten Notebook. Jika berhasil, layar *output* akan menghasilkan teks sesuai yang diharapkan di bawah Konten Notebook seperti pada Gambar 6.6. Setelah berhasil, kita siap untuk membuat program Python menggunakan Google Colab ini.



Gambar 6.6 Hasil Keluaran Contoh Sederhana Kode Program Python
Sumber: Dokumen Kemendikbud, 2021

2. Python

Ayo Mulai

Bahasa Pemrograman Python

Aktivitas AD-K10-02-P: Mengenal Python

Aktivitas pemanasan ini dilakukan untuk memulai mengenal sebuah bahasa pemrograman yang menyediakan *library* untuk analisis data, yaitu Python.

Untuk membantu berkomunikasi dengan komputer, kita perlu sebuah bahasa yang dipahami baik oleh manusia maupun komputer, dalam hal ini bahasa Pemrograman. Secara umum, bahasa pemrograman adalah bahasa yang digunakan untuk membuat program yang akan memberikan perintah kepada komputer untuk melakukan sesuatu. Ada berbagai jenis bahasa pemrograman yang dapat digunakan untuk membuat program. Salah satu bahasa pemrograman yang sering digunakan untuk melakukan analisis data karena menyediakan *library* untuk berbagai tahap proses analisis data, ialah bahasa Python. Python merupakan bahasa pemrograman yang cukup populer, seperti halnya bahasa C yang digunakan dalam unit pembelajaran Algoritma dan Pemrograman. Pada bagian ini, akan dibahas beberapa dasar pemrograman Python, terutama yang terkait dengan analisis data. Perhatikan bahwa pemrograman di materi analisis data ini hanya digunakan sebagai alat bantu. Algoritma dan pemrograman juga akan sedikit disinggung dalam unit pembelajaran ini sebagai bentuk latihan dan eksplorasi, memahami eksekusi yang terjadi khususnya dalam setiap proses analisis data. Jadi, kalian tidak perlu terlalu bingung memikirkan aturan bahasa pemrograman Python yang diberikan.

a. Cetak Data (*print*)

Perintah yang paling sederhana dalam bahasa pemrograman ialah perintah untuk mencetak suatu nilai atau data (*output*). Seperti halnya pada Gambar 6.6 yang sudah disajikan sebelumnya, mencetak data dalam Python dapat

dilakukan dengan menggunakan perintah `print`. Kalian dapat mengubah parameter atau isi di dalam tanda kurung `print` dengan data yang lain, misalnya menggantinya dengan “Selamat Datang”. Setelah kode program dieksekusi atau dengan menekan tombol *Run*, program akan menampilkan data seperti yang diharapkan.

Latihan 1

1. Gantilah data dalam *print* dengan `print(“2 + 3”)`, kemudian jalankan (*Run*). Apa hasilnya?
2. Kemudian, ganti kembali data dalam `print` dengan `print(2 + 3)`. Apa hasilnya?
3. Ganti kembali data dalam *print* dengan `print(“2” + “3”)`. Apa hasilnya?
4. Apa makna dari penggunaan tanda petik tersebut?

Tanda petik digunakan untuk mencetak data *string*, yang berupa nilai alfanumerik dan tanda baca, yang tidak memiliki nilai aritmatika. Artinya, *string* “2” ditambah (atau lebih tepatnya dijejerkan) dengan *string* “3”, hasilnya *string* “23”. Tidak masuk akal string “2” dikurangi atau dikalikan *string* “3”, sedangkan jika bilangan 2 dikurangi atau dikalikan 3, pasti ada nilai bilangannya. Sampai di sini, apakah kalian paham? Silakan, diskusikan dengan teman dan guru kalian.

Pemrograman Python di sini serupa dengan pemrograman C di unit pembelajaran Algoritma Pemrograman. Di program-program selanjutnya, kalian akan belajar Python dengan pendekatan prosedural seperti halnya belajar C. Untuk melihat hubungannya, kalian bisa melihat kembali tabel perbandingan bahasa Python dan bahasa C di unit pembelajaran Algoritma Pemrograman.

b. Pemberian Nilai Data (Assignment)

Selain mencetak langsung dari data yang ada di dalam tanda kurung, data lain yang pengisiannya di luar tanda kurung juga dapat dicetak. Pengisian atau pemberian nilai data ini disebut *assignment. Assignment* dilakukan dengan menyediakan sebuah nama (variabel) yang kemudian diisi dengan suatu nilai data menggunakan tanda sama dengan (“=”). Perhatikan baris perintah pertama pada Gambar 6.7. Dalam baris tersebut, variabel bil1 diisi dengan nilai 10. Sebelum ke baris keempat, perhatikan catatan berikut.

```
bil1 = 10
bil2 = 5
jumlah = bil1 + bil2
jumlah
```

Gambar 6.7 Contoh Assignment dalam Operasi Penjumlahan Sederhana
Sumber: Dokumen Kemendikbud, 2021

Latihan 2

1. Jika diperhatikan, ada berapa nama variabel yang digunakan dalam Gambar 6.6?
2. Apa makna dari baris ke-2 dan ke-3 dalam Gambar 6.6?

Setelah dilakukan penjumlahan dan data nilai hasilnya disimpan ke dalam variabel jumlah, data ini kemudian dicetak pada baris 4. Kita dapat melihat hasil cetak atau keluaran (output) dari variabel jumlah setelah program ditulis lengkap kemudian di-*Run*. Perhatikan bahwa mencetak dapat dilakukan cukup dengan menuliskan nama variabel yang akan dicetak tanpa menggunakan perintah *print*, khususnya jika berada di baris terakhir kode program.

Latihan 3

Sebelum perintah menjumlahkan (baris 3), tambahkan baris perintah untuk mencetak bil2 tanpa menggunakan perintah *print*.

```
bil1 = 10
bil2 = 5
bil2
jumlah = bil1 + bil2
jumlah
```

1. Dapatkah `bil2` tercetak di hasil eksekusi?
2. Kemudian, sekarang tambahkan perintah *print* pada baris untuk mencetak `bil2` dan `jumlah`.

```
bil1 = 10
bil2 = 5
print(bil2)
jumlah = bil1 + bil2
print(jumlah)
```

Dapatkah kedua variabel tersebut tercetak?

c. Banyak Nilai untuk Satu Variabel (*Array*)

Latihan 4

1. Ketikkan kode program berikut, kemudian lihat hasil eksekusinya.

```
bil1 = 10
print(bil1)
bil1 = 5
print(bil1)
```

2. Apakah *output* dari variabel `bil1` pada baris keempat? Nilai 5 saja? Ke manakah nilai 10-nya?

Perhatikan bahwa pada variabel `bil1`, `bil2`, dan `jumlah`, kita menyimpan data dengan satu nilai saja untuk setiap variabel. Jika kita mengisi kembali variabel dengan nilai yang lain, nilai yang lama akan hilang atau ditimpa dengan nilai baru. Namun demikian, kita masih bisa menyimpan lebih dari satu nilai dalam satu variabel saja, yang disebut variabel array. Dalam kehidupan sehari-hari, kita akan menemui data yang memiliki banyak nilai, termasuk daftar nilai mata pelajaran Informatika sebuah kelas, misalnya. Menuliskan sebuah *array* dapat dilakukan dengan menggunakan tanda kurung siku seperti diperlihatkan pada Gambar 6.8 baris pertama.

```
data = [10, 9, 7, 8, 10, 8]
print(data[1])
data[2] = 10
print(data)
data.append(9)
print(data)
```

Gambar 6.8 Hasil Keluaran Contoh Sederhana Kode Program Python
Sumber: Dokumen Kemendikbud, 2021

Karena satu variabel menyimpan banyak nilai, kita dapat menyebutkan nilai-nilai tersebut sebagai nilai pertama (atau elemen indeks pertama), elemen kedua, elemen ketiga, dan seterusnya. Dalam bahasa pemrograman, mengakses nilai ke-n dari suatu array dapat dilakukan dengan menggunakan kurung siku persis setelah nama *variabel array* tersebut, misalnya pada baris kedua Gambar 6.8.

Latihan 5

1. Ketikkan baris perintah **pertama** dan **kedua** saja pada kode program Gambar 6.7 (tanpa kode baris ketiga dan seterusnya), kemudian jalankan programnya. Apakah *output*-nya bilangan 9?
2. Padahal, kita ingin mencetak data ke-1 pada baris kedua tersebut. Mengapa tidak muncul bilangan 10?

Indeks array dalam bahasa Python atau di hampir semua bahasa pemrograman dimulai dengan perhitungan nol, disebut indeks ke-0. Bilangan 9 seperti yang tercetak dari Gambar 6.7 ialah elemen dengan indeks ke-1 atau indeks pertama.

d. Penelusuran Data/Pengulangan (*Loop: for*)

Perhatikan bahwa dalam Gambar 6.8 baris pertama, kita hanya mencetak satu elemen sebuah data array pada indeks pertama. Pada baris ketiga dan kelima, kita mencetak banyak elemen, tetapi sebagai satu data utuh. Bagaimana jika kita ingin mencetak array sebagai elemen-elemen terpisah? Misalnya, kita ingin mencetak setiap elemen dengan memberikan keterangan indeksnya. Kita dapat menggunakan konsep Perulangan atau *Loop* dengan menggunakan perintah **for**, seperti ditunjukkan Gambar 6.9. Perintah **for** digunakan dengan menyertakan sebuah variabel baru untuk menelusuri setiap elemen di dalam variabel yang diberikan (setelah perintah in). Perhatikan baris ketiga Gambar 6.9.

```
data = [10, 9, 7, 8, 10, 8]
indeks = 0
for elemen in data:
  print("Elemen ke ", indeks, " = ", elemen)
  indeks = indeks + 1
```

Gambar 6.9 Contoh Penggunaan for dalam Python
Sumber: Dokumen Kemendikbud, 2021

Karena for merupakan perintah blok (mengandung beberapa perintah lain), perlu diakhiri dengan tanda titik dua. Perintah-perintah yang dimaksudkan untuk ikut berulang mengikuti for ini dibuat menjorok ke kanan misalnya 1 tab. Perhatikan penulisan baris keempat dan kelima Gambar 6.9.

Latihan 6

1. Perhatikan baris kelima Gambar 6.9. Mengapa nomor indeks yang dicetak di baris sebelumnya, perlu ditambah 1 setiap perulangan?
2. Bagaimana jika diganti, indeks ditambah dengan 2? Apa keluarannya?
3. Bagaimana jika dipindah, penambahan indeks dilakukan di baris keempat? Apa *output*-nya?

Kita dapat menggunakan perintah lain untuk mengulang, seperti perintah **while**. Kita dapat bereksplorasi terkait penggunaan **while** ini.

Latihan 7

Cari tahu penggunaan **while** dalam bahasa Python!

e. Uji Kondisi (*Branch: try*)

Perhatikan bahwa kode program pada Gambar 6.9, *array* diisi dengan satu jenis data, yaitu data bilangan. Namun demikian, sebuah *array* bisa diisi dengan data yang bervariasi, misal data bilangan dan data tekstual (karakter dengan tanda petik). Namun, bagaimana pengolahan data yang bervariasi seperti ini?

Latihan 8

Ketik kode program pada Gambar 6.10! Kode program tersebut mencetak semua data di dalam *array* yang kemudian dibagi dua. Namun demikian, pengolahan seperti ini menghasilkan *error*.
Error apa yang muncul? Apa maknanya?

```
data = [10, 9, "Cindi", 4, "8"]
for x in data:
  print(x / 2)

5.0
4.5
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent
<ipython-input-5-39c5096a6524> in <module>()
      1 data = [10, 9, "Cindi", 4, "8"]
      2 for x in data:
----> 3   print(x / 2)

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Gambar 6.10 Contoh Pengolahan Data yang Bervariasi
Sumber: Dokumen Kemendikbud, 2021

Perhatikan bahwa data yang dicetak dengan dibagi dua terlebih dahulu ini memiliki variasi data seperti data bilangan dan data tekstual. Padahal, pembagian hanya bisa dilakukan pada bilangan dan oleh bilangan. Maka, akan terjadi kesalahan jika pembagian dilakukan pada data bilangan dan data tekstual. Perhatikan pesan kesalahannya berbunyi "*Line 3. Unsupported operand types* **for** /", yang dibagi harus berupa bilangan dengan bilangan, bukan string/teks.

Lalu, bagaimana jika kondisi data yang akan diolah bervariasi seperti di atas? Ada kalanya, *user* atau pengguna program memasukkan data yang benar. Namun ada kalanya, mereka melakukan kesalahan saat memasukkan data. Oleh karena itu, program yang dibuat harus memiliki mekanisme untuk menangani variasi data tersebut. Bagaimana penanganannya? Misalnya, jika data yang ada berupa bilangan, data langsung dibagi. Namun, jika data tidak bisa dibagi, berikan pesan bahwa data bukan merupakan angka/bilangan. Penanganan ini dalam konsep pemrograman disebut konsep percabangan atau branch. Salah satu perintah dalam Python yang dapat digunakan dalam percabangan ialah pasangan perintah **try** dan **except** seperti diperlihatkan pada Gambar 6.11.

```python
data = [10, 9, "Cindi", 4, "8"]
for x in data:
  try:
    print(x / 2)
  except:
    print("Bukan bilangan")
```

Gambar 6.11 Contoh Penggunaan try-except dalam Python
Sumber: Dokumen Kemendikbud, 2021

Secara umum, penggunaan blok **try-except** dapat dijelaskan dengan terjemahannya. Kita mencoba (trying) untuk mengeksekusi perintah-perintah (baris setelah **try).** Jika terjadi kesalahan, eksekusi akan dilempar/lompat ke baris except dan mengeksekusi baris di dalamnya. Jika tidak terjadi kesalahan hingga **akhir** perintah dalam **try**, akan keluar dari blok perintah **try-except** (dalam hal ini kembali ke perintah **for** untuk perulangan berikutnya). Dari mekanisme ini, muncul proses percabangan di mana ada pemrosesan data yang membagi bilangan dan mencetak hasil pembagiannya, dan ada pemrosesan data yang tidak mencetak bilangan, tetapi mencetak pesan "Bukan bilangan".

Latihan 9
Ketik kode program pada Gambar 6.11! Berapa bilangan yang tercetak?

Blok perintah **try-except** merupakan salah satu perintah yang digunakan untuk kasus percabangan untuk menguji suatu kondisi, dalam hal ini kasus kesalahan atau tidak. Ada perintah lain yang dapat digunakan untuk menangani kasus percabangan (*branch*), seperti perintah **if** dan blok perintah **if-else.**

Latihan 10
Cari tahu penggunaan branch if atau **if-else** dalam bahasa Python!

f. Pustaka Kode (*Library import*)

Saat memerlukan informasi yang mungkin belum pernah dipelajari dan tidak ditemukan di buku mata pelajaran yang kalian miliki, kalian akan pergi ke perpustakaan yang menyimpan banyak sekali buku dan informasi yang dapat kita cari. Seperti halnya perpustakaan sekolah, fitur *library* atau pustaka di sebuah pemrograman memungkinkan kita untuk mendapatkan fungsionalitas yang tidak ada di program kita. Fungsionalitas ini disediakan oleh penyedia bahasa pemrograman atau komunitas tertentu yang memang sering menggunakan fungsionalitas tersebut. *Library* atau pustaka adalah koleksi program dan paket yang tersedia untuk berbagai penggunaan.

Sebagai contoh, terdapat *library* **Pandas.** Kalian dapat mencoba untuk mengetikkan kode program seperti pada Gambar 6.12 sehingga menampilkan keluaran seperti pada Gambar 6.13. Perhatikan bahwa sebelum *library* **Pandas** digunakan di Baris 6 (**pandas.DataFrame**), kita perlu **import** terlebih dahulu *library* tersebut di Baris 1. *Library* **Pandas** ini dapat kalian pelajari lebih detil dari halaman webnya di https://pandas.pydata.org/.

```
import pandas

data = [10, 9, 7, 8, 10]
siswa = ["Andi", "Budi", "Cindi", "Dedy", "Edi"]

nilai = pandas.DataFrame({
    "Nama": siswa,
    "Nilai": data
})
nilai
```

Gambar 6.12 Contoh sederhana penggunaan import pustaka Pandas
Sumber: Dokumen Kemendikbud, 2021

	Nama	Nilai
0	Andi	10
1	Budi	9
2	Cindi	7
3	Dedy	8
4	Edi	10

Gambar 6.13 Hasil keluaran contoh sederhana penggunaan import pustaka Pandas
Sumber: Dokumen Kemendikbud, 2021

B. Koleksi Data



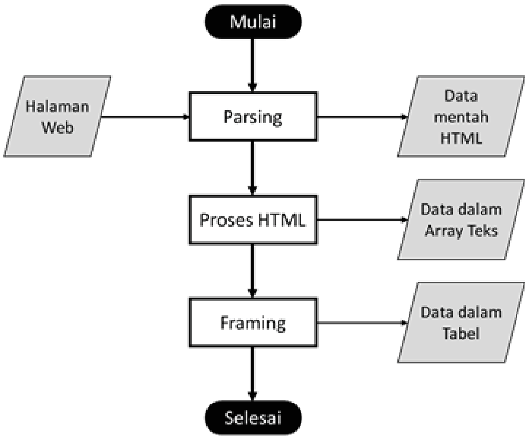
Web Scraping

Aktivitas AD-K10-03-P: Proyek *Web Scraping*

Aktivitas inti dilakukan untuk mengenal proses analisis data, khususnya dalam hal mengoleksi data dari situs web, yang dikenal dengan istilah web *scraping*.

Gambar 6.14 menunjukkan beberapa langkah yang bisa diikuti untuk melakukan *scraping* dalam bahasa Python menggunakan editor *online* Google Colab. Setelah editor siap digunakan, *scraping* bisa dimulai dengan proses parsing. Parsing adalah mengambil kode program dari sebuah halaman website secara utuh yang masih dalam bentuk kode HTML. Selanjutnya, kode HTML tersebut diproses setiap elemennya untuk mendapatkan data yang penting yang akan dirangkum. Hasilnya berupa kumpulan data yang diperlukan saja (yang diambil dari data mentah HTML yang utuh sebelumnya). Hasil keluaran sebelumnya masih berupa daftar atau *array* teks data pekerjaan yang

mungkin masih sulit dibaca. Data tersebut perlu ditampilkan secara lebih tertata sehingga mudah dibaca. Proses membingkai data atau *framing* ini bisa dilakukan dengan mudah jika data sudah diperoleh. Salah satu tampilan yang memudahkan pembacaan daftar teks tersebut ialah dalam bentuk tabel. Dengan demikian, kita peroleh hasil dari proses *scraping* ini dalam bentuk tabel data.



Gambar 6.14 Alur Proses Web Scraping
Sumber: Dokumen Kemendikbud, 2021

Mari, ikuti pembuatan proyek *scraping* tersebut dengan mengikuti langkah-langkah berikut.

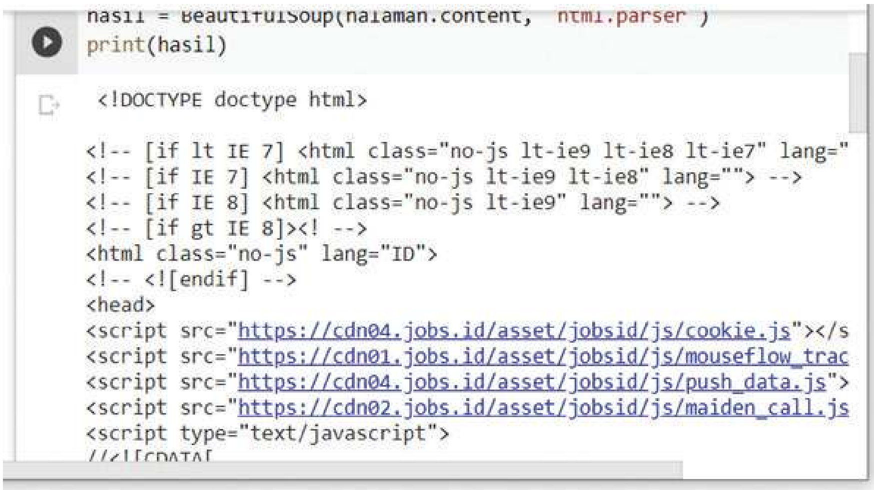
1. Buka Google Colaboratory melalui link https://colab.research.google.com/. Kemudian, buka catatan baru melalui menu File > New notebook.
2. Parsing salah satu alamat website lowongan pekerjaan. Gambar 6.15 menunjukkan kode untuk melakukan parsing alamat website Lowongan Pekerjaan yang digunakan sebelumnya. Ketik beberapa baris kode program berikut di layar Google Colab kalian.

```
import requests
import pandas as pd
from bs4 import BeautifulSoup

th = "https://www.jobs.id/lowongan-kerja?kata-kunci=part time"
halaman = requests.get(th)
hasil = BeautifulSoup(halaman.content, 'html.parser')
print(hasil)
```

Gambar 6.15 Kode Program untuk Parsing Sebuah Alamat Web
Sumber: Dokumen Kemendikbud, 2021

Jika dijalankan, kode tersebut akan menyalin kode program yang ada di alamat website yang di-*request* seperti dicontohkan pada Gambar 6.16.



Gambar 6.16 Hasil Keluaran dari Kode Program Parsing Website Lowongan Pekerjaan
Sumber: Dokumen Kemendikbud, 2021

3. Olah kode HTML tersebut hingga kalian bisa mengambil data posisi pekerjaannya, instansi yang memberikan pekerjaan, serta gaji yang ditawarkan seperti diperlihatkan Gambar 6.17. Tiga data ini bisa diperoleh dari elemen kode website yang disebut *tag*. Setiap tiga elemen tersebut tersimpan di *tag* tertentu untuk kemudian diambil seperti diperlihatkan pada Gambar 6.17. Lanjutkan kode program yang ada pada Gambar 6.15 sehingga lengkap seperti pada Gambar 6.17.

```
import requests
import pandas as pd
from bs4 import BeautifulSoup

th = "https://www.jobs.id/lowongan-kerja?kata-kunci=part time"
halaman = requests.get(th)
hasil = BeautifulSoup(halaman.content, 'html.parser')
lowkers = hasil.find_all(class_="single-job-ads")

posisi = []
instansi = []
gaji = []

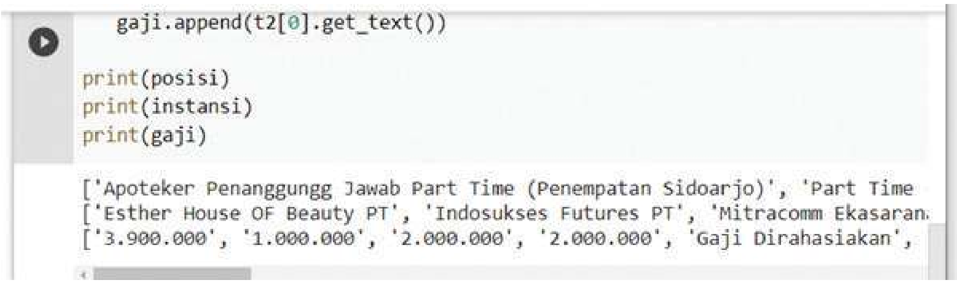
for p in lowkers:
 t1 = p.select("h3")
 t2 = t1[0].select("a")
 posisi.append(t2[0].get_text())

 t1 = p.select("p")
 t2 = t1[0].select("a")
```

```
21   try:
22     instansi.append(t2[0].get_text())
23   except:
24     instansi.append("-")
25
26   t2 = t1[1].select("span")
27   try:
28     gaji.append(t2[1].get_text())
29   except:
30     gaji.append(t2[0].get_text())
31
32 print(posisi)
33 print(instansi)
34 print(gaji)
```

Gambar 6.17 Kode Program untuk Mengoleksi Data Lowongan Pekerjaan
Sumber: Dokumen Kemendikbud, 2021

Jika kode program tersebut dijalankan di Google Colab, keluaran yang dihasilkan berupa ringkasan data dari yang diharapkan saja, seperti diperlihatkan pada Gambar 6.18.



Gambar 6.18 Hasil Keluaran dari Kode Program Koleksi Data Pekerjaan
Sumber: Dokumen Kemendikbud, 2021

4. Terakhir, bingkai data array tersebut ke dalam tabel yang ditampilkan oleh Python seperti diperlihatkan kode pada Gambar 6.19. Ketikkan kode program tersebut menggantikan baris 32-34 yang sudah diketik pada Gambar 6.17.

```
31
32 lowker = pd.DataFrame({
33     "Posisi": posisi,
34     "Instansi": instansi,
35     "Gaji":gaji
36 })
37 lowker
```

Gambar 6.19 Kode Program untuk Membingkai Data ke dalam Tabel
Sumber: Dokumen Kemendikbud, 2021

```
    "Gaji":gaji
})
lowker
```

	Posisi	Instansi	Gaji
0	Apoteker Penanggungg Jawab Part Time (Penempat...	Esther House OF Beauty PT	3.900.000
1	Part Time - Corporate Investment Consultant	Indosukses Futures PT	1.000.000
2	Customer Care Part Time	Mitracomm Ekasarana PT	2.000.000
3	Call Center Officer Part Time	Mitracomm Ekasarana PT	2.000.000
4	CALL Center PART TIME - Jakarta	Mitracomm Ekasarana PT	Gaji Dirahasiakan
5	Deputy Division Head (Production Assistant Man...	Indo Tirta Abadi PT	Gaji Dirahasiakan

Gambar 6.20 Contoh Hasil Keluaran dari Kode Program yang Menampilkan Tabel
Sumber: Dokumen Kemendikbud, 2021

Jika kode program tersebut dijalankan di Google Colab, keluaran yang dihasilkan berupa tabel dari data yang telah dikoleksi, seperti diperlihatkan pada Gambar 6.20.

Gambar 6.20. menunjukkan hasil akhir proses *scraping* data dari website Jobs ID untuk mengoleksi data lowongan pekerjaan yang meringkasnya ke dalam sebuah tabel. Bagaimana kalian membaca lowongan pekerjaan dari Gambar 6.1 sebagai halaman web asli dan Gambar 6.20 sebagai hasil *scraping*? Dengan penyajian seperti pada Gambar 6.20, data yang diperoleh akan lebih mudah dipahami dan kemudian diolah/dianalisis. Perhatikan bahwa tampilan tersebut merupakan data saat dibuat. Bisa jadi, keluaran tidak sama persis, bahkan mungkin juga terjadi *error.*

Langkah-langkah *scraping* menggunakan bahasa Python seperti ini dapat dipelajari secara lebih detail dari tutorial yang ada di internet seperti misalnya di https://www.dataquest.io/blog/web-scraping-tutorial-python/. *Scraping* juga dapat dilakukan menggunakan bahasa lain (PHP atau R-Language) dan editor (*Integrated Development Environment*) lain.

C. Visualisasi Data



Aktivitas AD-K10-04-P: Proyek Visualisasi Data

Aktivitas lanjutan ini dilakukan untuk mengenal visualisasi data.

Pada proyek sebelumnya, data website diambil (*scraping*) dan ditampilkan dalam bentuk tabel. Tabel adalah salah satu bentuk analisis data dasar. Kita bisa melihat persebaran data secara baris per baris dalam bentuk tekstual. Analisis data dapat dilanjutkan dengan mengubah data tekstual tersebut menjadi data visual sehingga lebih mudah untuk dipahami. Data divisualkan

dalam berbagai diagram seperti diagram batang, diagram lingkaran, diagram garis, dan lain sebagainya. Berikut ini data pada proyek sebelumnya akan disajikan dalam diagram batang.

Sebelum data bisa diolah dan disajikan menjadi diagram, data tersebut perlu dipersiapkan terlebih dahulu. Dalam analisis data, proses persiapan ini disebut pra-pemrosesan data (*data preprocessing*). Setelah dipersiapkan, baru kemudian data bisa diolah dan divisualisasikan.

1. Pra-pemrosesan Data

Pra-pemrosesan data dilakukan agar data siap untuk diolah. Data perlu dipersiapkan karena bisa jadi yang kita peroleh masih mentah, banyak terdapat kesalahan sehingga tidak bisa dihitung untuk dibuat visualisasinya. Sebagai contoh, pada kasus sebelumnya, data yang diolah, yaitu data gaji, masih ada yang berupa data teks seperti “Dirahasiakan”. Gaji juga masih mengandung titik yang mengelompokkan tiga angka. Padahal, dalam bilangan (integer), titik bermakna pecahan. Oleh karena itu, kita perlu mengubah data gaji yang masih bertipe teks ini ke dalam tipe bilangan seperti diperlihatkan pada Gambar 6.21. Baris 27-34 berikut perlu menggantikan kode program Baris 27-30 pada Gambar 6.17.

```
26    t2 = t1[1].select(“span”)
27    try:
28     xgaji = t2[1].get_text()
29    except:
30     xgaji = t2[0].get_text()
31    xgaji = xgaji.replace(“.”,””)
32    if (xgaji==”Gaji Dirahasiakan”):
33     xgaji = 0
34    gaji.append(xgaji);
```

Gambar 6.21 Kode Program untuk Preprocessing dalam Menata Data Gaji
Sumber: Dokumen Kemendikbud, 2021

2. Visualisasi Data: Barchart

Untuk membuat tampilan grafik atau diagram, kita dapat menggunakan *library* Python, yaitu Plotly. *Library* ini perlu ditambahkan terlebih dahulu di kode program. Cara penambahannya menggunakan **import** seperti yang sudah dipelajari sebelumnya, seperti berikut.

```
import plotly.express as px
```

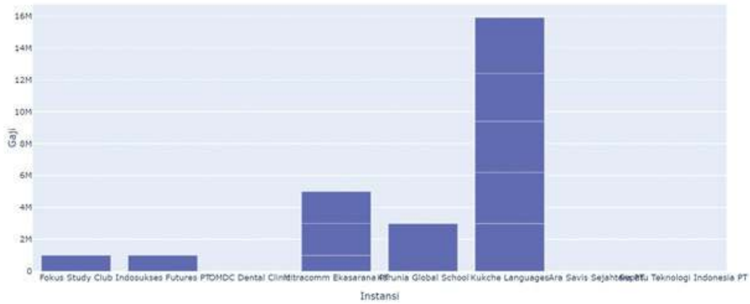
Selanjutnya, setelah *library*-nya di-import, kita dapat menggunakan fungsionalitas pada *library* Plotly untuk membuat diagram. Sebagai contoh, kita dapat membuat diagram batang (barchart) menggunakan fungsi bar() seperti diperlihatkan pada Gambar 6.22.

```
39
40  lowker = pd.DataFrame({
41    "Posisi": posisi,
42    "Instansi": instansi,
43    "Gaji":gaji
44  })
45
46  fig = px.bar(lowker, x='Instansi', y='Gaji')
47  fig.show()
```

Gambar 6.22 Kode Program untuk Memvisualisasi Data dalam Diagram Batang

Sumber: Dokumen Kemendikbud, 2021

Dari kode program tersebut, kita akan menghasilkan sebuah diagram batang seperti yang dicontohkan pada Gambar 6.23.



Gambar 6.23 Contoh Hasil Keluaran dari Kode Program Visualisasi Data

Sumber: Dokumen Kemendikbud, 2021

3. Prapemrosesan Data Lanjut

Jika diperhatikan pada Gambar 6.23, beberapa lowongan pekerjaan terlihat menumpuk. Beberapa posisi lowongan pada satu instansi menjadi satu batang saja. Hal ini menyebabkan nominal gaji yang ditunjukkan menjadi tidak valid. Ini terjadi karena kita mengelompokkan lowongan berdasarkan instansinya (Gambar 6.22 Baris 46). Bagaimana jika dikelompokkan berdasarkan posisinya? Tentunya, hal ini bukan menjadi solusi karena bisa jadi terdapat beberapa lowongan dengan posisi yang sama dan akan tertumpuk juga. Jadi, bagaimana solusinya?

Solusinya ialah mengelompokkan lowongan pekerjaan berdasarkan gabungan posisi dan instansi penyedia lowongan. Oleh karena itu, kita perlu mengubah kembali kode pada Gambar 6.17 sehingga posisi dan instansi tidak berada dalam variabel yang berbeda, tetapi disatukan dalam satu variabel (dalam hal ini variabel posisi saja) seperti diperlihatkan pada Gambar 6.24 Baris 18-28 yang menggantikan kode program sebelumnya. Selanjutnya, bingkai data tersebut ditampilkan cukup untuk dua item saja, yaitu posisi dan gaji seperti pada Baris 41-42. Terakhir, untuk menampilkan diagram batangnya, item yang digunakan diganti dari instansi menjadi posisi seperti pada Baris 45 Gambar 6.24 (menggantikan Baris 46 Gambar 6.22). Dengan demikian, setiap lowongan akan diperlihatkan secara terpisah seperti pada Gambar 6.25.

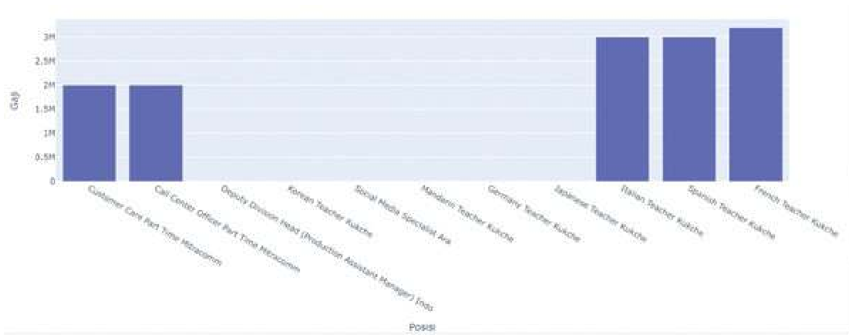
```
 t1 = p.select("h3")
 t2 = t1[0].select("a")
 t3 = t2[0].get_text()

 t1 = p.select("p")
 t2 = t1[0].select("a")
 try:
   t2 = t2[0].get_text()
 except:
   t2 = ""
 if (t2.find(" ")>=0):
   t3 = t3 + " " + t2[0:t2.find(" ")]
 posisi.append(t3)

lowker = pd.DataFrame({
   "Posisi": posisi,
   "Gaji":gaji
})

fig = px.bar(lowker, x='Posisi', y='Gaji')
fig.show()
```

Gambar 6.24 Kode Program untuk *Prcproccssing* dalam Mengelompokkan Posisi dan Institusi
Sumber: Dokumen Kemendikbud, 2021



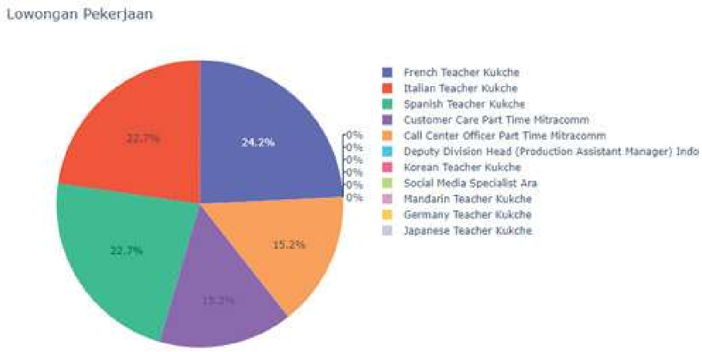
Gambar 6.25 Contoh Hasil Keluaran Setelah Posisi dan Institusi Dikelompokkan
Sumber: Dokumen Kemendikbud, 2021

4. Visualisasi Data dengan Diagram lain

Setiap diagram dan *chart* memiliki kegunaan masing-masing sesuai karakteristiknya. Diagram batang, misalnya, digunakan untuk memperlihatkan beberapa item serupa yang perlu dibandingkan. Di sisi lain, diagram lingkaran digunakan untuk memperlihatkan proporsi dari beberapa item yang bisa menjadi bagian utuh dari suatu kasus. Untuk membuat diagram lainnya, kalian dapat mempelajari penggunaanya di alamat ini https://plotly.

com/python/basic-charts/ Sebagai contoh, dapat dicoba untuk membuat diagram lingkaran (*pie chart*) seperti diperlihatkan pada Gambar 6.26. Untuk melakukannya, kode program yang sebelumnya menampilkan diagram batang (Gambar 6.24 Baris 45) diganti menjadi kode untuk menampilkan diagram lingkaran seperti berikut.

```
fig = px.pie(lowker, values='Gaji', names='Posisi', title='Lowongan')
```



Gambar 6.26 Contoh Hasil Keluaran dari Kode Program Visualisasi Data Menggunakan Diagram Lingkaran
Sumber: Dokumen Kemendikbud, 2021

Menarik ,bukan? Bagaimana dengan diagram yang lain?

Latihan 11

Untuk membuat diagram *scatter* (diagram titik) dari data lowongan pekerjaan ini, bagaimana kodenya?

Sebagai catatan, sebenarnya, diagram lingkaran dan diagram scatter kurang tepat jika digunakan dalam kasus ini. Namun demikian, kita dapat menggunakannya sebagai contoh untuk dipelajari pembuatannya karena datanya sudah tersedia. Dapatkah kalian mencari contoh kasus lain yang tepat divisualisasikan menggunakan kedua diagram tersebut?

Ayo Kita Renungkan

Pada unit pembelajaran ini, kalian telah mempelajari tentang bagaimana mengimplementasi sebuah *web scraper* dengan menggunakan *library* dari bahasa Python. Hal ini menjadi alternatif cara yang menggantikan cara sebelumnya, yaitu membuka secara manual satu per satu halaman webnya. Program web scraper ini akan melakukannya untuk kalian. Dari sini, ternyata, kalian telah melakukan aktivitas pemrograman seperti halnya yang dilakukan pada unit Algoritma Pemrograman.

Tantangan Berpikir

Jika kalian perhatikan, apa bedanya aktivitas pemrograman yang dilakukan di unit Analisis Data ini dibandingkan dengan yang dilakukan di unit Algoritma Pemrograman? Pertanyaan renungan berikutnya, apa bedanya mesin scraper yang kalian buat dan mesin pencari yang sering kalian gunakan?

Target *Scraping*

Kita telah melakukan web *scraping* untuk web yang terbuka dan dapat dilihat oleh publik. Jadi, program scraper, seperti halnya seseorang yang secara manual melakukannya, akan dapat membuka suatu halaman web karena informasi yang ditampilkan pada halaman web tersebut ialah informasi publik. Bagaimana dengan website yang tidak publik? Misalnya, kita hanya dapat menampilkan email setelah melakukan login. Informasi email kita bukan merupakan informasi publik, melainkan informasi privat. Saat kita menampilkan halaman kita di sosial media, hanya kita dan teman-teman kita yang dapat melihat. Halaman di sosial media ini tidak publik dan juga tidak privat, tetapi terbatas kepada sekumpulan tertentu. Apakah kita bisa membuat scraper untuk informasi yang tidak publik? Bolehkah kita mengambil data yang privat seperti itu? Diskusikan dengan teman-teman!

Tentang Perkakas

Kalian telah memanfaatkan *library* Python untuk melakukan *scraping*. Program *scraper* yang kalian tulis hanya untuk "membungkus" *library* tersebut agar dapat dijalankan. Menurut kalian, bagaimana program *scraper* melakukan pengambilan isi halaman web?

Library untuk *scraping* hanya merupakan salah satu *library* Python untuk analisis data secara keseluruhan. Python tidak hanya menyediakan *library* untuk *scraping*, tetapi juga menyediakan banyak *library* untuk keperluan lainnya. Jika kalian membutuhkan untuk keperluan lain, kalian akan dapat memanfaatkan seperti kalian memanfaatkan *library* untuk *scraping*. Eksplorasilah, *library* apa lagi yang dimiliki Python?

Sistem Visualisasi Real-Time

Saat kalian melakukan pemesanan ojek *online*, pelanggan dapat melacak posisi pengendara atau *driver* sedang di mana saat ini (Gambar 6.27.a). Pada beberapa tempat misalnya di bandara (Gambar 6.27.b), di halte bus atau di tempat umum, seringkali ada tampilan yang selalu diubah sesuai dengan keadaan saat itu. Pada sistem penerbangan di bandara, misalnya, setiap kali ada pesawat berangkat atau pergi, tampilan akan diubah. Kemudian di bank, tampilan kurs mata uang hari ini juga selalu diperbarui karena kurs tersebut sering berubah (Gambar 6.27.c). Data kurs ini diambil dari sistem komputer Bank Indonesia (Bank Sentral).

Berbagai data yang sering berubah tersebut menunjukkan bahwa yang ditampilkan ialah data pada waktu saat itu juga atau yang disebut real-time. Visualisasi data real-time ini dapat dilihat pada Gambar 6.26. Untuk kepentingan yang lebih besar, data *real-time* ini divisualisasikan dalam sistem yang disebut *dashboard*. Jika ingin belajar lebih jauh tentang sistem *dashboard*, kalian dapat mempelajari *website* https://dash-gallery.plotly.host/dash-web-trader/.

Rangkaian program komputer dibutuhkan dari berbagai sumber data. Program komputer tersebut akan berkolaborasi untuk akhirnya menampilkan data yang dilihat pengguna. Bayangkan bahwa setiap mesin/agen pengolah tersebut adalah sebuah program komputer yang berfungsi sesuai peran masing-masing, seperti manusia yang bersinergi di dunia dunia nyata. Namun, dalam hal ini, program komputer merupakan program yang aktif bekerja, bukan di dunia fisikal, tetapi di dunia digital. Dapatkah kalian membayangkan hal ini? Buatlah ilustrasi kerja sama yang dilakukan oleh para agen pengolah tersebut! Buatlah infografis dalam bentuk diagram untuk 3 sistem yang diilustrasikan pada Gambar 6.26.



Gambar 6.27 Contoh Visualisasi Data Secara *Real-Time*

Ada satu catatan menarik mengenai aplikasi ojek *online* yang bisa didiskusikan. Jika kalian pernah melakukan pemesanan ojek *online*, kalian lihat bahwa sepeda motor bergerak, tetapi tidak lancar. Mengapa ini bisa terjadi?

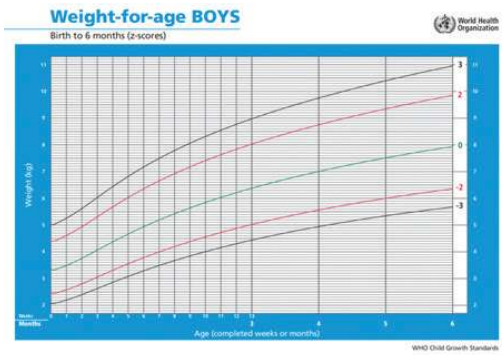
Kembali ke proyek web *scraping*, program scraper ini hanya merupakan sebagian kecil dari rangkaian analisis data. Rangkaian yang lengkap ialah mengambil, memroses, menyajikan dalam visualisasi yang memudahkan pembaca, menganalisis, menginterpretasi, dan menyimpulkan atau mengambil keputusan. Dari ilustrasi rangkaian proses analisis data tersebut, kita dapat ketahui bahwa pengambilan data dapat dilakukan dengan program, misalnya dengan program yang telah kalian buat.

Selanjutnya, dari rangkaian proses tersebut, menurut kalian, apakah semuanya dapat diotomasi dengan menulis programnya? Proses mana yang dapat digantikan oleh program komputer, dan mana yang sulit atau bahkan tidak mungkin dijadikan program komputer? Kalaupun dapat digantikan oleh program komputer, apakah memang lebih menguntungkan untuk membuat

programnya? Jelaskan alasan kalian!

Tentang Prediksi

Prediksi dari data dapat dilakukan dengan mudah jika divisualisasikan. Misalnya, dokter dapat menggunakan kurva pertumbuhan berat badan bayi untuk memprediksi perkembangan bayi dan menyatakan kesehatannya (Gambar 6.28).



Gambar 6.28 Visualisasi untuk Data Berat Badan sebagai Alat Memprediksi Pertumbuhan Ideal Bayi

Dengan pengalaman melakukan web *scraping* tentang lowongan pekerjaan, kalian dapat mencoba untuk membuat grafik mengenai tren kebutuhan *programmer*. Misalnya, kalian dapat mengambil data lowongan tahun 2019 dan tahun 2020, atau data lowongan setiap bulan di tahun 2021. Selalu menaik, bukan? Dari data tersebut, kalian dapat memperkirakan kira-kira berapa lowongan pekerjaan yang ada di bulan depan.

Uji Kompetensi

1. Jelaskan langkah koleksi data/*scraping* dengan bahasa kalian sendiri!
2. Proyek *web scraping* sebelumnya mencontohkan salah satu lowongan pekerjaan, yaitu "*part-time*". Bagian mana yang harus diubah agar scraping tersebut menampilkan lowongan pekerjaan lain, misalnya "*programmer*"?
3. Proyek *web scraping* sebelumnya menampilkan informasi posisi, instansi, dan gaji suatu lowongan pekerjaan yang ditampilkan dalam tiga kolom tabel. Dapatkan kalian menambah satu informasi lagi, yaitu Lokasi ke dalam tabel lowongan pekerjaan tersebut? Bagaimana analisis data untuk struktur HTML website lowongan pekerjaan tersebut?
4. Cari lowongan pekerjaan yang terdapat di koran/majalah (dilakukan secara manual/*unplugged*) atau *website* (dilakukan dengan bantuan komputer/*plugged*)! Rangkum seperti tabel yang dicontohkan pada Aktivitas 3! Tabel dapat ditulis tangan dalam selembar kertas (*unplugged*) atau ditulis di Aplikasi Pengolah Angka (*plugged*). Proses apa saja yang kalian lakukan jika analisis data dilakukan secara manual, bukan otomatis seperti pada aktivitas *web scraping*?

5. Cari lowongan pekerjaan dari *website* lain, misalnya JobsDB yang berada di alamat https://id.jobsdb.com/id yang memiliki struktur HTML yang sederhana dan mudah dianalisis. Dapatkah kalian melakukan *scraping* dari website tersebut untuk mendapatkan rangkuman lowongan pekerjaan yang tersedia dalam sebuah tabel? Kalian dapat menggunakan tabel berikut untuk membantu pekerjaan kalian.

No	Aktivitas	Checklist (√)	Keterangan Output	Jawaban
1.	Parsing HTML Website	▯	Berapa baris kode HTML yang diperoleh?	
2.	Proses struktur HTML	▯	Berapa *array* informasi dari setiap lowongan yang diperoleh?	
3.	*Framing* data	▯	Berapa baris lowongan pekerjaan dari tabel yang ditampilkan?	

Ingin Tahu Lebih?

Jika kalian tertarik dengan materi ini dan ingin mempelajari lebih lanjut, kalian dapat mengakses ke link berikut ini:

1. Jobs ID (2020). Info Lowongan Kerja Terbaru dan Populer 2020. Diakses dari http://jobs.id
2. Wikipedia (2020). Web Scraping. Diakses dari https://en.wikipedia.org/wiki/Web_scraping
3. Google Colaboratory (2020). Welcome to Colaboratory. Diakses dari https://colab.research.google.com/
4. Dataquest (2020). Tutorial: Web Scraping with Python using Beautiful Soup. Diakses dari https://www.dataquest.io/blog/web-scraping-tutorial-python/
5. PyData (2021). Pandas: Python Data Analysis *Library*. Diakses dari https://pandas.pydata.org/
6. Ploty (2021). Plotly Python Open Source Graphing *Library* Basic Chart. Diakses dati https://plotly.com/python/basic-charts/
7. Plotly (2021). Plotly | Dash. Diakses dari https://dash-gallery.plotly.host/dash-web-trader/

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Dean Apriana Ramadhan, Auzi Asfarian, Mushthofa
ISBN : 978-602-244-506-7

Bab 7
Algoritma dan Pemrograman



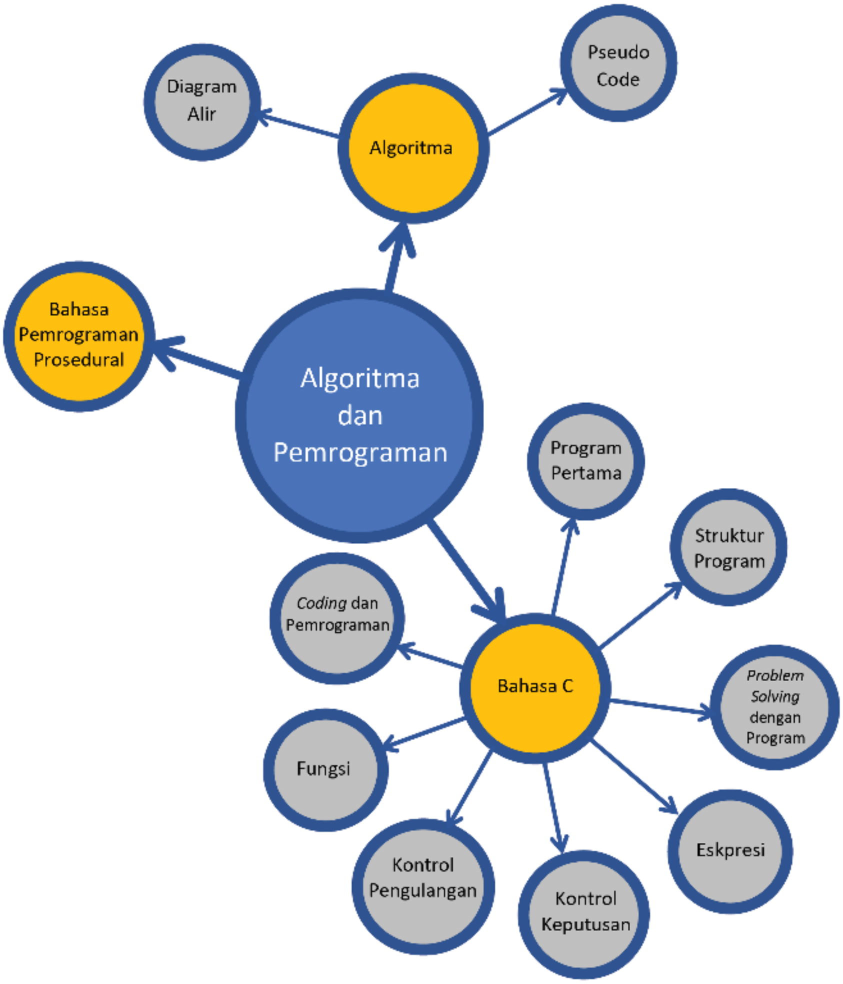
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu (a) membaca dan menulis algoritma dengan notasi yang benar, memahami proses pemrograman dengan menggunakan bahasa pemrograman; (b) memahami konsep variabel dan ekspresi dalam membuat program; (c) memahami penggunaan struktur kontrol keputusan, struktur kontrol perulangan, dan fungsi dalam membuat program.

Pertanyaan Pemantik

Kalian ingin dapat memprogram dengan bahasa pemrograman? Bagaimana membuat program untuk membuat solusi-solusi kecil?

Peta Konsep



Gambar 7.1 Peta Konsep Algoritma dan Pemrograman

Apersepsi

Selama ini, mungkin kalian telah menggunakan banyak sekali produk perangkat lunak baik di komputer personal maupun di ponsel pintar yang kalian gunakan. Sekarang saatnya bagi kalian untuk membuat program sendiri. Pada bagian ini, kalian akan berkenalan dengan konsep algoritma dan pemrograman. Untuk membuat program, kalian harus menggunakan salah satu dari banyak bahasa pemrograman yang ada.

Kata Kunci

Penyelesaian Persoalan (*problem solving*), Algoritma, pemrograman, *coding, debugging.*

A. Algoritma

Algoritma adalah suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan dalam bentuk program komputer untuk menyelesaikan suatu permasalahan komputasi tertentu. Algoritma merupakan bentuk dari suatu strategi atau 'resep' yang kalian gunakan untuk menyelesaikan suatu masalah. Algoritma lahir dari suatu proses berpikir komputasional oleh seseorang untuk menemukan solusi dari suatu permasalahan yang diberikan. Dengan demikian, berpikir komputasional merupakan keahlian yang kalian perlukan untuk dapat membuat algoritma, program, atau suatu karya informatika yang dapat digunakan dengan efektif dan efisien.

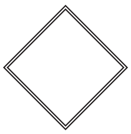
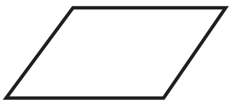
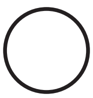
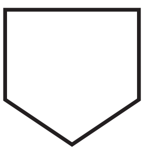
Setelah kalian menganalisis suatu problem menggunakan teknik abstraksi dan dekomposisi lalu menyusun algoritma dengan melakukan pengenalan pola dari problem sejenis, algoritma tersebut harus direpresentasikan dalam bentuk yang dapat dipahami oleh orang lain. Selain itu, karena pada akhirnya strategi tersebut akan diubah dalam bentuk kode program, algoritma harus ditulis dalam bentuk yang terdefinisi dengan baik (*well-defined*) dengan jumlah langkah yang terbatas. Algoritma adalah abstraksi dari sebuah program sehingga kemampuan menuliskan algoritma dengan baik akan membantu kalian dalam membuat program yang baik dan benar.

Pada bagian ini, kalian akan mempelajari dua cara untuk merepresentasikan algoritma, yaitu diagram alir dan *pseudocode*. Untuk itu, kalian perlu mempelajari teknik untuk membaca suatu algoritma (yang disebut penelusuran atau *tracing*) dan cara untuk menuliskan suatu algoritma. Perlu diingat bahwa menulis algoritma berbeda dengan menulis program. Program ditulis agar dapat dipahami oleh mesin, sedangkan algoritma ditulis agar dapat dipahami oleh manusia. Untuk program yang sederhana, algoritma akan sangat mirip, bahkan sama dengan program. Jika persoalan makin kompleks, algoritma hanya berisi abstraksi, yang akan mempermudah implementasinya menjadi program.

1. Diagram Alir

Diagram alir dibuat dalam bentuk aliran simbol yang dapat ditelusuri dari suatu titik permulaan hingga titik akhir dari program. Diagram alir dibuat menggunakan simbol standar ANSI/ISO yang beberapa simbol dasarnya diberikan pada Tabel 7.1.

Tabel 7.1 Simbol Diagram Alir Beserta Maknanya

Simbol	Nama	Deskripsi
	Garis alir (*flowline*)	Arah yang menunjukkan aliran program dari awal hingga akhir.
	Terminator	Titik awal atau titik akhir suatu program.
	Proses	Suatu kegiatan komputasi yang dilakukan oleh program: misalnya operasi aritmatika.
	Keputusan	Merupakan titik percabangan yang salah satu cabangnya dapat dilalui oleh program berdasarkan suatu kondisi.
	Masukan (*Input*)/Keluaran (*Output*)	Melambangkan titik saat program akan menerima suatu data atau menghasilkan suatu informasi.
	Subprogram	Melambangkan suatu kegiatan atau proses lain yang telah didefinisikan sebelumnya.
	Penghubung dalam Halaman	Digunakan untuk menghubungkan suatu titik pada diagram alir ke titik lain pada halaman yang sama.
	Penghubung antarhalaman	Digunakan untuk menghubungkan suatu titik pada diagram alir ke titik lain pada halaman yang berbeda. Digunakan apabila diagram lain cukup kompleks sehingga tidak dapat digambar dalam satu halaman.

Untuk memahami bagaimana diagram alir digunakan untuk menggambarkan suatu algoritma, pada bagian berikut, diberikan lima buah contoh diagram alir dari beberapa proses berpikir yang telah kalian kenal.

a. Diagram Alir 1: Menghitung Luas Persegi

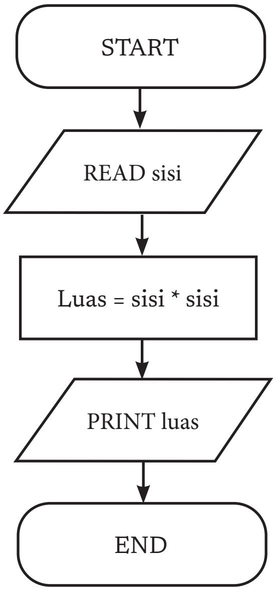


Diagram alir dibaca mulai dari simbol START, lalu mengikuti arah panah.

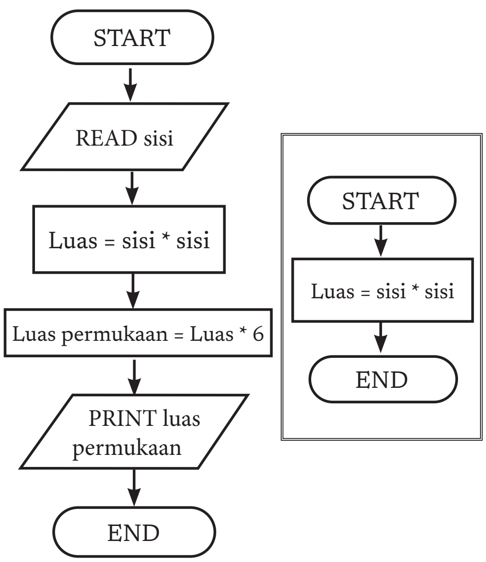
Untuk menghitung luas persegi, kalian memerlukan sebuah data, yaitu panjang sisi. Panjang sisi ini dibaca pada diagram alir dengan menggunakan kata kunci *READ* dan disimpan dalam sebuah variabel bernama sisi.

Setelah itu, kalian melakukan suatu proses ekspresi matematika untuk menghitung luas persegi menggunakan rumus yang telah kalian ketahui, yaitu luas = sisi x sisi. Hasil perhitungan tersebut disimpan pada sebuah variabel bernama luas.

Walaupun telah memperoleh jawaban yang dicari, komputer perlu diinstruksikan secara spesifik untuk mengeluarkan jawaban tersebut. Pada diagram ini, kata kunci *PRINT* digunakan untuk mencetak nilai dari variabel luas yang telah diperoleh pada tahap sebelumnya.

Setelah *PRINT*, algoritma berakhir karena simbol setelahnya adalah simbol *END*.

b. Diagram Alir 2: Menghitung Luas Permukaan Kubus



Saat menyusun solusi untuk menyelesaikan suatu permasalahan, kita seringkali membutuhkan solusi dari permasalahan lain yang lebih sederhana.

Misalnya, kita harus menghitung luas permukaan dari sebuah kubus. Pada prosesnya, kita perlu menghitung luas persegi yang membentuk kubus tersebut. Hal ini dapat digambarkan pada diagram alir menggunakan simbol subprogram.

Pada diagram alir ini, terlihat bahwa proses akan memanggil subprogram menghitung luas persegi yang telah kita buat sebelumnya.

Subprogram dapat kalian gunakan untuk menggambarkan abstraksi dan dekomposisi yang telah kalian pelajari pada berpikir komputasional.

c. Diagram Alir 3: Membagi Bilangan

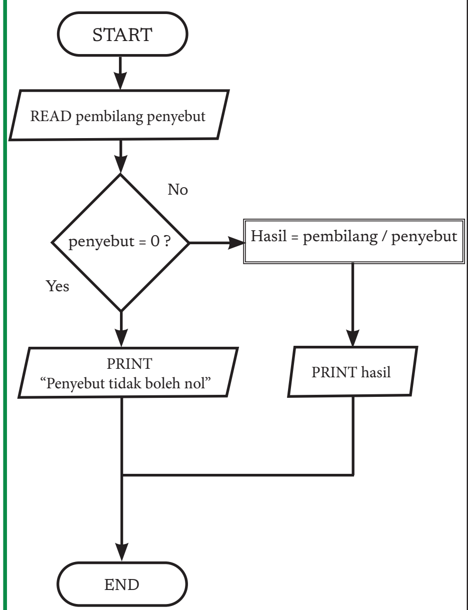
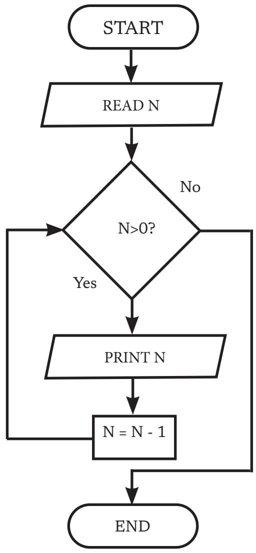


Diagram alir dapat memiliki beberapa kemungkinan aliran sehingga suatu algoritma dapat adaptif terhadap masukan yang diberikan. Hal ini dimungkinkan dengan adanya simbol keputusan. Aliran keluar dari simbol keputusan akan bergantung pada kondisi yang ada di dalam simbol keputusan.

Pada contoh ini, simbol keputusan digunakan untuk menghindari dijalankannya suatu operasi matematika yang tidak dapat dieksekusi oleh komputer, yaitu operasi pembagian dengan pembagi bernilai 0. Apabila operasi tersebut dilakukan, komputer akan menampilkan pesan kesalahan dan program akan berhenti secara tidak wajar.

Diagram alir ini merupakan proses untuk membagi pembilang dengan penyebut. Akan tetapi, sebelum operasi pembagian dilakukan, diagram akan mengecek terlebih dahulu nilai dari penyebut. Apabila penyebut bernilai 0, operasi pembagian tidak dilakukan dan pesan yang sesuai akan ditampilkan. Jika tidak, operasi dapat dilakukan dengan aman dan hasil pembagian dapat ditampilkan.

d. Diagram Alir 4: Menghitung Mundur dari N hingga 1

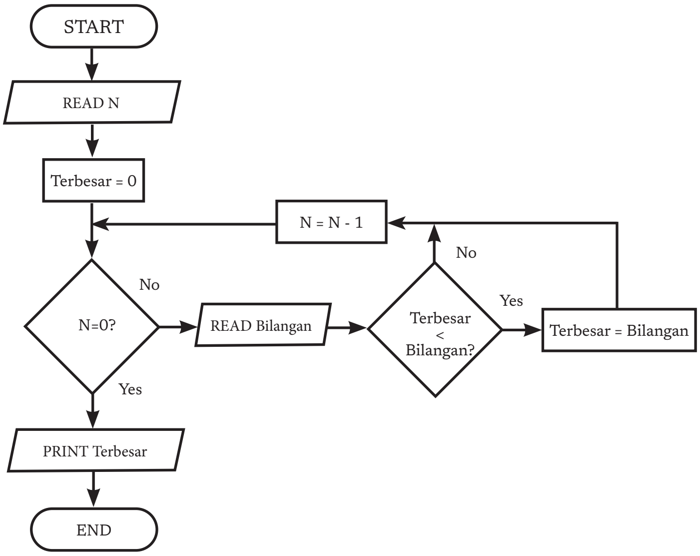


Aliran pada diagram alir dapat diatur sehingga satu lebih simbol dijalankan berulang kali. Pada contoh berikut, perulangan dilakukan sehingga diagram alir tersebut akan menghasilkan barisan bilangan bulat dari N hingga 1. Misalkan, N bernilai 5. Maka, diagram alir akan mencetak angka 5 4 3 2 1.

Tentunya, perulangan tidak bisa dilakukan terus-menerus sehingga diperlukan suatu kondisi untuk menghentikan perulangan. Simbol keputusan dapat digunakan untuk menghentikan perulangan tersebut pada kondisi yang kita tetapkan.

e. Diagram Alir 5: Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan

Tentunya, simbol-simbol dasar pada diagram alir dapat dipadukan untuk menghasilkan sebuah proses yang lebih kompleks. Diagram alir berikut menggambarkan proses mencari bilangan terbesar dari suatu himpunan bilangan yang diberikan. Diagram alir berikut akan membaca sebanyak N buah bilangan dan akan menghasilkan bilangan yang paling besar di antara bilangan tersebut.



f. Menelusuri Diagram Alir

Di buku kerja kalian, kalian dapat melakukan penelusuran (*tracing*) secara terstruktur dengan membuat tabel sederhana yang terbagi tiga bagian, yaitu masukan, nilai variabel, dan keluaran. Bagian masukan akan diisi dengan data yang akan diproses, bagian nilai variabel akan menjadi tempat kalian mencatat nilai yang disimpan di dalam variabel, sedangkan bagian keluaran akan diisi dengan data yang dihasilkan oleh diagram alir. Penelusuran ini menjadi penting saat kalian ingin memahami perilaku dari suatu algoritma, atau saat kalian ingin mengecek ketepatan keluaran dari suatu algoritma.

Sebagai contoh, misal kalian mencari bilangan terbesar dari 4 bilangan berikut: 1, 3, 2, 4 menggunakan diagram alir kelima. Ada beberapa kegiatan inti yang akan kalian lakukan dalam melakukan penelusuran.

1. Mempersiapkan lembar kerja penelusuran (Tabel 7.2.a), kalian menuliskan data yang akan diolah, menuliskan nama variabel yang digunakan pada diagram alir, dan mengosongkan bagian keluaran.
2. Setelah itu, kalian mulai menelusuri diagram alir dari bagian permulaan.

3. Ketika menemukan simbol untuk membaca suatu data, kalian dapat mencoret masukan yang dibaca, kemudian meletakkannya ke variabel yang tepat. Misal, kalian berada pada simbol masukan READ N. Letakkan bilangan pertama di bagian masukan (yaitu 4) ke variabel N (Tabel 7.2.b).
4. Ketika kalian tiba di suatu simbol proses yang menyimpan suatu nilai pada variabel (penugasan atau *assignment*), kalian meletakkan nilai tersebut pada bagian nilai variabel. Misalnya, saat bertemu dengan Terbesar = 0 pada diagram alir, kalian menuliskan nilai 0 pada variabel terbesar (Tabel 7.2.c).
5. Proses juga dapat berisi ekspresi matematika, misalnya N = N - 1. Untuk mengerjakan ekspresi tersebut, kerjakan dahulu bagian kanan dari ekspresi, yaitu N - 1. Cek nilai N saat ini di bagian nilai variabel, dan kalian akan menemukan nilai 4. Kerjakan ekspresi tersebut, yaitu 4 - 1 = 3, lalu simpan hasilnya ke sisi kanan dari ekspresi yaitu variabel N. Lewat ekspresi ini, nilai N yang tadinya 4 sekarang telah berubah menjadi 3. Kalian dapat mencoret nilai 4 dan menuliskan nilai 3 pada Lembar Kerja (Tabel 7.2.d).
6. Kemudian, kalian dapat melanjut proses dan akhirnya menemukan nilai N sekarang bernilai 0. Lembar Kerja kalian akan berisi seperti Tabel 7.2.e.
7. Kemudian, kalian menemukan simbol keluaran PRINT Terbesar. Pada tahap ini, kalian dapat menuliskan isi dari variabel terbesar ke dalam bagian keluaran di lembar kerja kalian (Tabel 7.2.f).
8. Terakhir, kalian menelusuri dan menemukan terminator END sehingga penelusuran berakhir. Dengan demikian, diagram alir tadi menghasilkan keluaran berupa nilai 4 pada kasus yang diberikan. Selamat! Kalian telah berhasil menelusuri diagram alir!

Tabel 7.2 Ilustrasi Penelusuran Diagram Alir di Buku Kerja Siswa.

(a) Sebelum kalian menelusuri:	(b) Saat menemukan simbol masukan dengan perintah READ N.	(c) Saat menemukan simbol proses dengan perintah terbesar = 0.
Masukan 4 1 3 2 4	Masukan 4 1 3 2 4	Masukan 4 1 3 2 4
Nilai variabel N: Bilangan: Terbesar:	Nilai variabel N: 4 Bilangan: Terbesar:	Nilai variabel N: 4 Bilangan: Terbesar: 0
Keluaran	Keluaran	Keluaran

(d) Saat menemukan simbol proses dengan ekspresi *N = N - 1.*	(e) Saat N bernilai 0, Lembar Kerja kalian akan menjadi seperti ini.	(c) Saat menemukan simbol proses dengan perintah terbesar = 0.
Masukan 4 1 3 2 4	Masukan 4 ~~1324~~	Masukan 4 ~~1 3 2 4~~
Nilai variabel N: 4 3 Bilangan: Terbesar:	Nilai variabel N: ~~4321~~ 0 Bilangan: ~~1324~~ 0 Terbesar: 4	Nilai variabel N: ~~4321~~ 0 Bilangan: ~~1324~~ 0 Terbesar: 4
Keluaran	Keluaran	Keluaran 4

Walaupun ilustrasi ini terdiri atas beberapa tabel, kalian cukup bekerja dengan satu tabel saat menelusuri diagram alir. Teknik penelusuran ini tidak hanya dapat digunakan untuk membaca suatu diagram alir, tetapi juga dapat digunakan untuk membaca pseudocode atau kode program. Untuk algoritma yang pendek, kalian mungkin tidak membutuhkan lembar kerja seperti ini. Namun, lembar kerja ini akan sangat bermanfaat ketika menelusuri suatu algoritma yang panjang dan kompleks. Sekarang, saatnya kalian berlatih menggunakan teknik ini!

Ayo Lakukan

Aktivitas Individu

Aktivitas AP-K10-01-U: Menelusuri Diagram Alir

Sebelum mencoba membuat diagram alir sendiri, kalian perlu kemampuan untuk membaca diagram alir yang telah tersedia. Oleh karena itu, pada latihan ini, kalian akan mencoba menelusuri diagram alir yang telah tersedia di atas ketika diberikan suatu kasus untuk diselesaikan. Walaupun kalian telah mengetahui apa hasilnya, lakukanlah penelusuran secara terstruktur mulai dari awal hingga akhir diagram alir.

1. Hitunglah luas permukaan sebuah kubus yang memiliki panjang sisi 20 cm dengan menggunakan diagram alir pada contoh kedua.
2. Hitunglah dua buah operasi pembagian berikut dengan menggunakan diagram alir pada contoh kedua. Pertama, pembagi bernilai 10 dan penyebut bernilai 2. Kedua, pembagi bernilai 8 dan penyebut bernilai 0.
3. Lakukan hitung mundur dari angka 5 hingga 1 dengan menelusuri diagram alir keempat.

Carilah bilangan terbesar dari 7 bilangan berikut: 3, 1, 2, 4, 5, 7, dan 9 dengan menggunakan diagram alir kelima.

2. Pseudocode

Pseudocode (kode semu atau kode pseudo) adalah suatu bahasa buatan manusia yang sifatnya informal untuk merepresentasikan algoritma. *Pseudocode* dibuat untuk menutupi kekurangan diagram alir dalam merepresentasikan konsep-konsep pemrograman terstruktur. *Pseudocode* memungkinkan representasi langkah-langkah yang lebih detail dan dekat dengan bahasa pemrograman. Karena sifatnya yang informal, tidak ada aturan khusus dalam standar notasi yang dapat digunakan. Akan tetapi, ada beberapa prinsip dasar yang perlu diperhatikan, yaitu satu baris untuk satu pernyataan (*statement*) dan pentingnya indentasi dalam menuliskan pernyataan. Indentasi ada untuk hierarki dari pernyataan. Misalnya, untuk menunjukkan bahwa suatu pernyataan merupakan bagian dari sebuah struktur kontrol keputusan atau struktur kontrol perulangan (lihat konsep blok pada pemrograman visual yang telah kalian pelajari di tingkat SMP).

Keempat diagram alir pada bagian sebelumnya dapat ditulis dalam bentuk *pseudocode* sebagai berikut.

a. *Pseudocode* 1: Menghitung Luas Persegi

Deskripsi tingkat tinggi:	Pseudocode:
1. Baca nilai sisi persegi. 2. Hitung luas dengan menguadratkan nilai sisinya. 3. Cetak luas.	*Algoritma Menghitung Luas Persegi* *Input: Nilai Panjang Sisi Persegi.* *Output: Luas Persegi tercetak.* *input sisi* *luas← sisi * sisi* *print luas*

b. *Pseudocode* 2: Menghitung Luas Permukaan Kubus

Deskripsi tingkat tinggi:	Pseudocode:
1. Baca nilai sisi kubus. 2. Hitung luas (persegi) dari sisi kubus. Kalikan luas kubus dengan angka 6 (banyaknya jumlah persegi pada kubus) untuk mendapatkan luas permukaan. 3. Cetak luas_permukaan.	*Algoritma Menghitung Luas Permukaan Kubus* *Input: Nilai Panjang Sisi Kubus.* *Output: Luas Permukaan Kubus tercetak.* *input sisi* *luas_permukaan ←luas(sisi) * 6* *print luas_permukaan*

c. *Pseudocode* 3: Membagi Bilangan

Deskripsi tingkat tinggi:	Pseudocode:
1. Baca nilai pembilang dan penyebut. 2. Jika penyebut bernilai 0, cetak tulisan "Penyebut tidak boleh nol". 3. Jika penyebut tidak bernilai nol, lakukan pembagian pembilang dengan penyebut dan simpan hasilnya. 4. Cetak hasil pembagian.	*Algoritma Membagi Bilangan* *Input: Pembilang dan Penyebut.* *Output: Hasil Pembagian tercetak.* *input pembilang, penyebut* *if penyebut = 0 print "Penyebut tidak boleh nol"* *else* *hasil ← pembilang / penyebut* *print hasil*

d. *Pseudocode* 4: Menghitung Mundur dari N hingga 1

Deskripsi tingkat tinggi:	Pseudocode:
1. Baca nilai N. 2. Selama N > 0, ulangi. 3. Cetak tulisan N. 4. Kurangi nilai N dengan 1.	*Algoritma Menghitung Mundur dari N hingga 1* *Input: Nilai N.* *Output: Angka hasil hitung mundur dari nilai N sampai 1 tercetak.* *input N* *while N > 0* *print N* *N ← N - 1*

e. *Pseudocode* 5: Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan

Deskripsi tingkat tinggi:	Pseudocode:
1. Jika himpunan bilangan kosong, maka tidak ada bilangan terbesar. 2. Jika himpunan bilangan tidak kosong, asumsikan bilangan pertama sebagai bilang terbesar saat ini. 3. Untuk setiap bilangan anggota himpunan: bandingkan bilangan tersebut dengan bilangan terbesar saat ini. Apabila bilangan tersebut lebih besar, maka bilangan tersebut akan menjadi bilangan terbesar saat ini.	*Algoritma Mencari Bilangan Terbesar* *Input: Himpunan bilangan L.* *Output: Bilangan terbesar pada himpunan bilangan L* *if size of (L)= 0 return null* *largest ← elemen pertama L* *for each item in L, do* *if item > largest, then* *largest ← item* *return largest*

Deskripsi tingkat tinggi:	Pseudocode:
4. Apabila langkah 3 telah dilakukan pada seluruh bilangan, bilangan terbesar saat ini akan menjadi bilangan terbesar di himpunan bilangan tersebut.	

Setelah selesai menyusun suatu algoritma, barulah suatu program dibuat dengan menggunakan bahasa pemrograman tertentu. Ada banyak bahasa yang dapat digunakan, misalnya bahasa C yang digunakan pada unit ini dan bahasa Python yang digunakan pada unit analisis data.

Ayo Lakukan

Aktivitas Individu

Aktivitas AP-K10-02-U: Menulis Algoritma

Pada latihan ini, kalian diminta untuk menuliskan suatu algoritma berdasarkan deskripsi berikut. Deskripsi ini memuat narasi tingkat tinggi dari algoritma yang perlu kalian buat dalam bentuk diagram alir dan *pseudocode*. Setelah selesai, kalian dapat menunjukkan hasil pekerjaan kalian kepada teman kalian untuk ditelusuri.

Soal 1: Membayar Bakso (Tingkat Kesulitan: ★★)

Buatlah sebuah diagram alir atau *pseudocode* dari proses berikut.



Sebuah mesin pembayaran otomatis dirancang untuk mampu menangani pembayaran pembelian bakso secara mandiri. Mesin ini mampu untuk memberikan kembalian dalam bentuk uang kertas atau uang logam. Mesin akan menerima dua buah masukan, yaitu total bayar dan jumlah uang yang dibayarkan oleh pelanggan. Apabila jumlah uang yang dibayarkan lebih besar atau sama dengan total bayar, mesin akan menghitung kembalian yang harus diberikan kepada pelanggan. Apabila terjadi sebaliknya, mesin akan menampilkan teks "Uang yang dibayarkan kurang".

Setelah diagram alir selesai, kalian dapat menelusurinya dengan menggunakan kasus berikut.

Kasus	Masukan	Keluaran
1	Total Bayar: 10000 Jumlah Uang: 15000	5000
2	Total Bayar: 20000 Jumlah Uang: 10000	Uang yang dibayarkan kurang.

Soal 2: Hadiah Bakso Gratis (Tingkat Kesulitan: ★★★)

Kalian adalah pengusaha bakso yang sukses. Agar usaha bakso kalian bisa lebih berkembang, kalian berencana untuk menambah sentuhan teknologi sehingga beberapa proses dapat berjalan secara otomatis. Inovasi yang kalian pikirkan ialah menggunakan sistem poin untuk memberikan diskon pada pelanggan. Poin ini akan diberikan pada saat pelanggan membayar di mesin pembayaran yang akan kalian buat. Setiap membayar, pelanggan akan menerima poin senilai harga bakso yang ia beli. Apabila total poin mencapai 100.000, pelanggan akan menerima satu porsi bakso gratis.

Kalian lalu memikirkan suatu proses berikut: setelah memesan bakso, pelanggan dapat membayar dengan menggunakan ponsel miliknya. Kemudian, mesin tersebut menambahkan total pembayaran ke total poin yang saat ini dimiliki oleh pelanggan. Apabila total poin yang dimiliki pelanggan lebih besar dari 100.000, mesin akan mengeluarkan kalimat "Anda mendapatkan kupon bakso gratis" dan mengurangi total poin pelanggan dengan nilai 100.000. Setelah itu, mesin akan menampilkan total poin pelanggan saat ini.

Setelah diagram alir selesai, kalian dapat menelusurinya dengan menggunakan kasus berikut.

Kasus	Masukan	Keluaran
1	Total Pembayaran: 80000 Total Poin Pelanggan Saat Ini: 10000	Poin Anda saat ini: 90000
2	Total Pembayaran: 20000 Total Poin Pelanggan Saat Ini: 90000	Anda mendapatkan kupon bakso gratis! Poin Anda saat ini: 10000

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal.

1. Apakah kalian merasa bahwa membuat algoritma dapat mempermudah kalian dalam menyelesaikan masalah?
2. Menurut kalian, kapan kalian akan menggunakan diagram alir dan kapan kalian akan menggunakan *pseudocode*? Mana yang paling mudah untuk kalian gunakan? Apa kelebihan dan kekurangan dari setiap pendekatan tersebut yang kalian rasakan?

3. Setelah ini, apa yang akan kalian lakukan untuk bisa membaca dan menulis algoritma dengan baik?
4. Pelajaran paling berkesan apa yang kalian dapatkan dari pertemuan ini?

B. Bahasa Pemrograman Prosedural

Belajar bahasa pemrograman sama halnya dengan belajar bahasa apa pun, dimulai dengan secara intuitif mengenal dan langsung memakai bahasa tersebut untuk keperluan sehari-hari yang penting sesuai kebutuhan, bukan dari teori bahasa. Seseorang dengan bahasa ibu bahasa Indonesia, saat belajar bahasa Inggris, akan mulai mengenal bahasa Inggris melalui "membaca" contoh-contoh kalimat sederhana yang sangat diperlukan dalam kehidupan sehari-hari, seperti mulai menyapa "Selamat pagi.", "Jam berapa?". Melalui contoh tersebut, ia akan belajar kosakata penting dan pola kalimat, misalnya kalimat pernyataan SPOK (Subjek, Predikat, Objek, Keterangan); struktur kalimat tanya, kalimat aklamasi, dan lain-lain. Selanjutnya, baru beranjak ke konsep yang lebih kompleks yang ada pada bahasa asing tersebut dan mengenal tata bahasa secara lebih formal dan mulai menulis.

Belajar pemrograman pada hakikatnya sama dengan belajar bahasa natural (bahasa manusia sehari-hari): seseorang belajar dari "membaca" program terlebih dulu, daripada "menulis" kode program. Proses menulis kode (*coding*) dapat dimulai setelah kalian membaca contoh-contoh program yang menjadi pola pembangun program kompleks. Bedanya dengan belajar bahasa natural, teks dalam bahasa pemrograman yang ditulis bukan dilafalkan dan dipahami sebagai teks "statis", melainkan juga dapat dipahami oleh mesin dan dapat dieksekusi (dijalankan).



Gambar 7.2 Elemen Generik dari Bahasa Pemrograman Prosedural
Sumber: Dokumen Kemendikbud, 2021

Terdapat banyak bahasa pemrograman, dan setiap bahasa memiliki paradigma, keunggulan, tantangan masing-masing. Pada unit ini, kalian diperkenalkan pada bahasa pemrograman C yang merupakan salah satu bahasa pemrograman prosedural. Saat mempelajari bahasa C pada unit ini di kelas X, kalian akan mempelajari empat elemen generik, yaitu variabel, ekspresi, struktur kontrol keputusan, dan struktur kontrol perulangan (Gambar 7.2). Empat elemen ini berlaku di semua bahasa pemrograman prosedural lainnya. Teks kode program dalam bahasa-bahasa pemrograman lain banyak yang mirip dengan teks bahasa C.

Oleh karena itu, kalian perlu menyadari bahwa unit ini tidak dibuat hanya agar kalian menguasai pemrograman dengan bahasa C, tetapi bagaimana kalian dapat menggunakan keempat elemen dasar tersebut dalam membuat suatu program.

C. Bahasa Pemrograman C

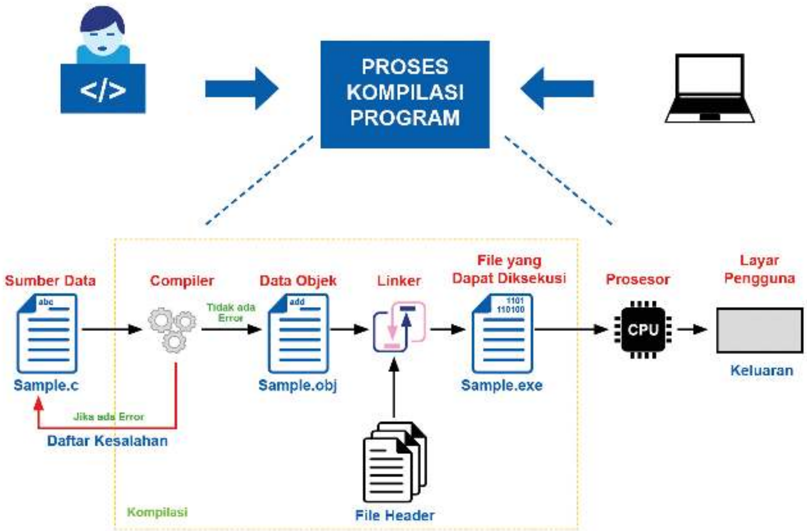
Bahasa Pemrograman C, selanjutnya disebut bahasa C saja, dikembangkan oleh Dennis M. Ritchie dan Brian W. Kernighan pada awal tahun 1970. Perkembangan bahasa C tidak bisa dipisahkan dari perkembangan sistem operasi UNIX, yang 90% lebih di antaranya ditulis dalam bahasa C. Karena sejarah yang panjang tersebut, kemudian muncul beberapa standar bahasa C yang merupakan spesifikasi dari bahasa C. Spesifikasi ini akan berpengaruh pada perilaku program yang dibuat dengan menggunakan bahasa tersebut. Beberapa standar yang ada meliputi: definisi Kernighan & Ritchie (K&R), ANSI-C (X-3.159 -1989-), Definisi AT&T (untuk superset C, C++), dan GNU Coding Standards. Versi pada PC misalnya: Lattice C, Microsoft C/Microsoft QuickC, dan Turbo C/Borland C++.

Bahasa C banyak dipakai untuk membuat sistem operasi dan program-program sistem, pemrograman tingkat rendah, atau yang "dekat" ke perangkat keras (misalnya untuk kontrol peralatan), membuat toolkit pemrograman, dan menulis aplikasi. Kelebihan bahasa C sehingga banyak digunakan ada pada kemampuannya untuk menghasilkan kode yang singkat, efisien, tetapi tetap mudah dibaca. Berbeda halnya dengan bahasa mesin yang efisien, tetapi membutuhkan latihan khusus untuk membacanya, atau bahasa tingkat tinggi lain yang enak dibaca, tetapi tidak efisien. Walaupun demikian, perlu diakui bahwa kesulitan untuk membaca program bahasa C lebih tinggi daripada bahasa tingkat tinggi lain.

1. Membuat Program Pertama dengan Bahasa C

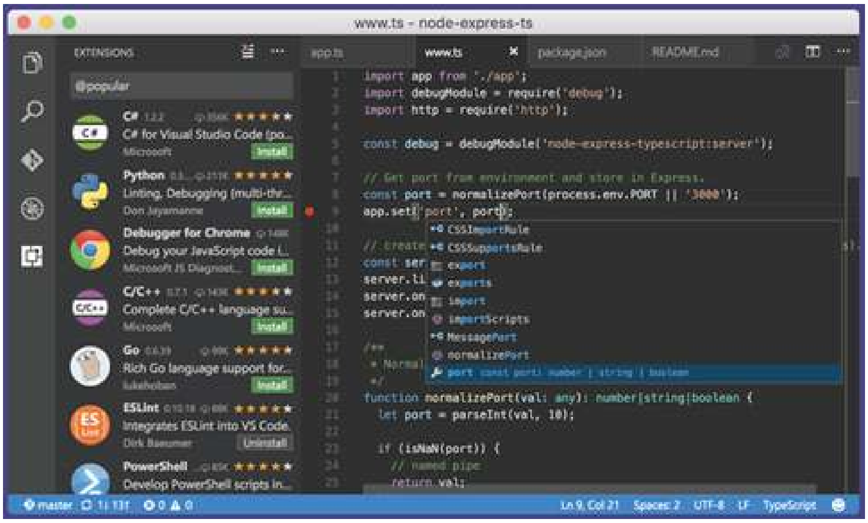
Sekarang, saatnya kalian memulai perjalanan kalian dalam membuat program dengan bahasa C. Namun, sebelum kalian mulai membuat program, ada persiapan yang harus kalian lakukan. Pertama, kalian membutuhkan sebuah tempat untuk bekerja, yang disebut lingkungan pengembangan. Kedua, kalian harus memahami proses membuat program mulai dari mengetikkan kode program hingga menghasilkan program yang dapat dieksekusi atau dijalankan oleh komputer.

a. Persiapan Lingkungan Pengembangan Program Menggunakan Bahasa C



Gambar 7.3 Alur proses membuat program dengan Bahasa C

Ini langkah-langkah untuk membuat program di C! (rumit sekali). Namun sekarang, ada IDE (*Integrated Development Environment*), semuanya jadi mudah! Tulis program, satu kali klik, langsung bisa tampil....



Gambar 7.4 Contoh tampilan sebuah IDE
Sumber: Dokumen Kemendikbud, 2021

Untuk dapat membuat program dalam bahasa C, diperlukan beberapa perangkat lunak. Perangkat lunak yang pertama ialah sebuah editor teks tempat kita mengetikkan kode program. Selanjutnya, ada sebuah kompilator (*compiler*) yang akan membaca kode bahasa C yang telah ditulis dan mengubahnya menjadi bahasa mesin, atau bahasa *assembly*. Setelah itu, terdapat sebuah assembler yang akan mengubah bahasa mesin tersebut ke dalam kode biner yang dapat dipahami dan dieksekusi oleh komputer. Terakhir, terdapat sebuah penghubung (*linker*) yang akan menyatukan beberapa berkas yang dihasilkan dalam proses-proses sebelumnya ke dalam sebuah bentuk berkas yang dapat dieksekusi (*executable*).

Pada awalnya, perangkat lunak tersebut terpisah, tetapi untuk memudahkan, akhirnya, dibuatlah sebuah perangkat lunak terintegrasi yang mencakup semua perangkat lunak di atas. Perangkat lunak tersebut disebut lingkungan pengembangan terpadu (*integrated development environment*). Untuk bahasa C, beberapa IDE yang biasa digunakan ialah Eclipse, Atom, Code::Blocks, Geany, dan Visual Studio. Walaupun pada dasarnya semua IDE tersebut memiliki fungsi yang sama, tetapi terdapat perbedaan pada fitur-fitur tambahan yang membuat proses pemrograman menjadi lebih mudah dilakukan. Misalnya, fitur untuk membuat program secara kolaboratif, integrasi dengan repositori kode program daring, serta fitur auto-complete. Selain IDE yang dipasang di komputer masing-masing, terdapat pula IDE yang terpasang di *cloud* dan dapat diakses secara daring, misalnya Ideone, tutorialspoint.com, dan onlinegdb.com. Selain itu, terdapat juga CppDroid, Mobile C, dan Coding C yang dapat digunakan pada ponsel cerdas.

Suatu perusahaan di bidang informatika biasanya telah memiliki standar masing-masing dari lingkungan pengembangan yang digunakan. Menguasai lingkungan pengembangan secara maksimal akan membuat kalian lebih produktif untuk menghasilkan program yang efisien dan berkualitas tinggi, terutama ketika kalian bekerja dalam tim.

Ayo Lakukan

Aktivitas Individu

Aktivitas AP-K10-03: Instalasi IDE Bahasa C

Sebelum membuat program, lingkungan kerja di komputer harus dipersiapkan. Pada bagian ini, akan dijelaskan cara instalasi IDE untuk memprogram bahasa C yang dapat digunakan secara gratis, yaitu Code::Blocks dan Geany. Cara instalasi yang diberikan ialah untuk dua sistem operasi, yaitu MS Windows dan Linux.

Sebelum lanjut, pastikan kalian telah mengetahui versi sistem operasi yang kalian gunakan, 32 bit atau 64 bit. Informasi ini dapat diperoleh dengan cara berikut.

1. Pada sistem operasi MS Windows: klik kanan ikon My Computer atau This PC, kemudian pilih Properties. Pada bagian System Type, terdapat informasi jumlah bit yang digunakan oleh komputer kalian.
2. Pada sistem operasi Linux: masuk ke terminal dan ketikkan perintah uname -a. Hal ini juga dapat dicek melalui antarmuka pengguna grafis pada bagian System Settings -> Details.

Ayo Lakukan

Aktivitas Individu

Aktivitas AP-K10-04-U: Membuat Program Pertama dengan Bahasa C

Pada aktivitas ini, kalian akan menggunakan Code::Blocks atau Geany untuk membuat program pertama kalian. Program yang akan ditulis ialah seperti berikut.

```
/* Program Pertamaku:                                    */
/* Membuat program untuk mencetak "Halo Dunia!"          */

#include<stdio.h>
int main() {
printf("Halo Dunia!\n");
return 0;
}
```

Ketikkanlah kode program tersebut ke dalam IDE yang kalian gunakan. Pada saat mengetik, kalian dapat mencoba memikirkan apa makna dari kode yang telah kalian ketikkan. Pastikan kalian mengetikkan semua kode dengan sempurna karena bahasa pemrograman sangat sensitif pada kesalahan pengetikan. Misalnya, ketika kalian lupa mengetikkan karakter ";" program kalian akan dianggap mengalami kesalahan sintaks oleh kompilator.

Petunjuk:

1. Pertama, buka Code::Blocks atau Geany. Kemudian, pilih File → New → Empty File.
2. Kedua, ketikkan kode yang diberikan di atas.
3. Ketiga, simpan berkas kode dengan nama hello_world.cpp. Simpan dalam sebuah direktori yang kalian buat di komputer. Namai berkas kode dan direktori dengan benar agar dapat kalian temukan dengan mudah nanti. Menyimpan kode dengan terstruktur merupakan salah satu praktik baik dalam menulis kode program.

4. Keempat, pilih Build → Build atau tekan Ctrl-F9 untuk melakukan kompilasi program. Pada tahap ini, kode program akan dikonversi ke dalam bentuk yang dapat dipahami dan dieksekusi oleh komputer.
5. Terakhir, pilih Build → Run atau tekan Ctrl-F10 untuk menjalankan program.

Setelah dikompilasi dan dijalankan, program ini akan mencetak keluaran ke layar monitor yang berupa sebaris teks bertuliskan:

```
Halo Dunia!
```

Selamat, kalian telah berhasil menjalankan program pertama kalian.

Praktik Baik Pemrograman

Sambil kalian mempelajari buku ini, akan ada sisipan praktik baik pemrograman. Praktik baik ini kalian terapkan dalam kegiatan pemrograman seterusnya, termasuk pada topik dan konsep yang belum tercakup dalam aktivitas ini, agar terkumpul dengan rapi. Kalian boleh menambahkan praktik baik lain yang kalian anggap perlu. Tuliskan praktik baik kalian dalam Buku Kerja kalian.

Pemrogram profesional harus menaati aturan-aturan yang lebih ketat. Beberapa perusahaan seperti Google bahkan mempunyai konvensi (kesepakatan) dan juga dikaitkan dengan bahasa pemrograman yang dipakai. Penasaran? Silakan lihat contohnya di google.github.io/styleguide.

Selama kelas X, kalian akan menuliskan banyak program kecil-kecil yang perlu kalian simpan untuk bahan belajar. Jika kalian perlukan, dapat kalian pakai lagi potongan kodenya untuk mempercepat kalian melakukan tugas yang lebih besar. Sebuah program besar biasanya berasal dari potongan-potongan program kecil yang pernah dibuat sebelumnya. Pemrogram tidak perlu mengetik ulang lagi karena potongan kode tersebut sudah pernah diuji. Program kecil ini ibarat potongan *puzzle* yang kelak dapat kalian ambil dan rangkai menjadi sesuatu yang besar.

Berikut ini praktik baik dalam memprogram serta mengelola direktori dan berkas tempat menyimpan kode program:.

1. Selalu menuliskan nama file tempat menyimpan potongan kode sebagai baris pertama kode, dalam bentuk komentar.
2. Menuliskan untuk apa potongan kode program ditulis, akan memudahkan mengambilnya tanpa harus membaca semua kode. Ingat bahwa sama halnya dengan belajar menulis, kalian akan menulis program yang ukurannya (banyaknya baris) makin lama makin besar.
3. Menuliskan komentar yang perlu untuk sekumpulan kode. Pada contoh program di atas, terdapat bagian komentar untuk menuliskan apa yang dilakukan oleh program. Komentar tidak akan dieksekusi oleh kompilator karena fungsinya untuk membantu kalian memahami kode program. Komentar dapat ditulis dalam dua cara.

a. Ditulis dalam satu baris atau lebih dan diawali dengan /* dan diakhiri dengan */.
b. Ditulis dalam satu baris dengan awalan //.

Kode program di atas dapat kalian berikan komentar seperti berikut.

```
/* Program Pertamaku:                                         */
/* Membuat program untuk mencetak "Halo Dunia!"               */

#include<stdio.h>
int main(){                         // Fungsi utama program
      printf("Halo Dunia!\n");      // Mencetak Halo Dunia! ke layar
      return 0;                     // Program berjalan dengan benar
}
```

Ayo Kita Renungkan

Jawablah pertanyaan tersebut dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal.

1. Apakah kalian sudah pernah membuat program sebelumnya?
2. Saat mengetikkan kode program tersebut, apakah kalian merasa dapat memahami makna dari setiap baris kode tersebut?
3. Apa kesulitan yang kalian rasakan saat mengetikkan program tersebut?
4. Pelajaran paling berkesan apa yang kalian dapatkan dari pertemuan ini?

2. Struktur Program Bahasa C

Bahasa C merupakan bahasa yang terstruktur. Beberapa struktur dasar pada program bahasa C telah terlihat pada program yang kalian tulis sebelumnya. Untuk memudahkan penjelasan, kode program berikut akan diberi nomor baris di bagian kiri.

Penting: Jika kalian ingin mencoba menjalankan contoh program dengan nomor baris seperti di bawah ini, kalian tidak perlu mengetikkan nomor baris tersebut.

```
#include<stdio.h>
int main(){
     printf("Halo Dunia!\n");
     return 0;
}
```

Baris pertama merupakan suatu pernyataan yang digunakan untuk memasukan sebuah *header file* yang bernama stdio.h. *Header file* merupakan kumpulan fungsi-fungsi dasar yang dikelompokkan berdasarkan kegunaannya

dan dapat digunakan untuk membantu kalian membuat program. Pada kode di atas, *header file* yang digunakan ialah stdio.h yang berisi fungsi-fungsi terkait masukan dan keluaran standar (atau standard input output). Dengan menggunakan stdio.h, program yang kalian buat dapat membaca dan menulis data.

Pada baris 2-5, kalian menemukan sebuah blok program yang merupakan implementasi suatu fungsi bernama main. Fungsi ini merupakan fungsi yang akan dijalankan pertama kali pada saat program dijalankan. Isi dari blok fungsi tersebut diapit dengan tanda kurung kurawal. Di dalam fungsi ini, pada baris 3, terdapat pemanggilan sebuah fungsi bernama `printf` yang digunakan untuk mencetak suatu data ke layar. Dalam hal ini, data yang ditampilkan ialah sebuah kalimat "Halo Dunia!\n" yang diberikan pada parameter fungsi tersebut. Karakter \n yang ada pada kalimat tersebut akan dicetak oleh program sebagai baris baru (*newline*), seperti jika kalian menekan tombol enter pada papan ketik kalian.

Pada baris keempat, terdapat sebuah pernyataan return 0. Penjelasan mengenai fungsi pernyataan ini cukup berat untuk dijelaskan pada bagian ini. Singkatnya, pernyataan ini merupakan tanda bagi sistem operasi untuk mengecek selesainya program dengan benar. Apabila program berjalan dan berhenti dengan benar, sistem operasi akan memperoleh nilai 0. Nilai selain 0 akan menjadi tanda bahwa program tidak berhenti dengan benar. Untuk saat ini, yang perlu kalian ketahui ialah pernyataan ini perlu diletakkan di akhir blok fungsi main pada program yang kalian tulis.

Kalian dapat memodifikasi kode program tersebut dan menjalankannya kembali dengan melakukan beberapa perubahan, misalnya:

1. Ubah kalimat Halo Dunia menjadi Hello World!
2. Cetak dua baris kalimat di layar, yaitu "Halo Dunia!" dan "Saya siap belajar pemrograman!"

a. Komponen Program dalam Bahasa C

Sebelum membuat program yang lebih kompleks, yang melibatkan ekspresi, struktur kontrol keputusan, atau struktur kontrol perulangan, kalian perlu memahami makna dari berbagai komponen penyusun bahasa C, di antaranya meliputi kata kunci, identifier, variabel, tipe data, dan konstanta.

1) Kata Kunci (*Keyword*)

Kata kunci merupakan kata yang telah memiliki makna khusus yang tidak dapat diubah oleh pemrograman. Dengan kata lain, kalian tidak dapat menggunakan sebagai suatu identifier. Terdapat 32 kata kunci standar pada bahasa C, yaitu:

`auto, double, int, struct, break, else, long, switch, case, enum, register, typedef, char, extern, return, union, continue, for, signed, void, do, if, static, while, default, goto, sizeof, volatile, const, float, short, dan unsigned.`

Bayangkan, dengan hanya 32 kata kunci, kalian bisa menulis program apa saja dalam bahasa C!

2) Identifier

Identifier adalah nama unik yang dapat kalian ingat yang diberikan ke dalam entitas program C, seperti variabel dan fungsi. Identifier terdiri atas serangkaian karakter dengan aturan berikut.

a. Tidak boleh sama dengan kata kunci (*keyword*) dalam bahasa C.
b. Disusun dari kombinasi huruf (besar dan kecil), angka, dan underscore "_".
c. Harus dimulai dengan huruf atau *underscore*.
d. Bersifat *case-sensitive*, atau sensitif terhadap huruf besar atau kecil (kapitalisasi karakter). Dengan kata lain, sisi dan Sisi akan dianggap sebagai dua identifier yang berbeda.

Praktik Baik Pemrograman

Walaupun dengan aturan di atas kalian dapat membuat *identifier* dengan sangat bebas, keterbacaan kode program menjadi penting. Beberapa praktik baik yang perlu kalian ketahui untuk menghindari kebingungan dalam membaca dan menulis kode ialah seperti berikut.

1. Nama yang diawali oleh underscore digunakan untuk keperluan tertentu dan tidak seharusnya digunakan dalam membuat program di unit ini.
2. Nama variabel dan fungsi harus ditulis dengan huruf non-kapital. Akan tetapi, apabila terdiri atas dua atau lebih kata, kalian dapat menggunakan teknik menulis dengan standar camel case. Huruf awal kata, selain kata pertama, ditulis menggunakan huruf kapital. Contoh: `totalHargaBarang, namaMahasiswa.`
3. Hindari menggunakan *identifier* yang sangat mirip dalam satu kode program. Misalnya, `totalHarga` dengan `total_harga`. Hindari juga dua atau lebih variabel yang hanya berbeda di kapitalisasi seperti `totalharga` dan `totalHarga.`
4. Hindari *identifier* yang terlihat mirip, misalnya karakter 'I', '1', dan 'l' terlihat sangat mirip satu sama lain. Catatan: karakter yang disebutkan ialah huruf i kapital, angka 1, dan huruf L non-kapital.
5. Identifier haruslah bersifat *mnemonic.* Untuk variabel, artinya identifier variabel tersebut harus menunjukkan isi dari variabel tersebut. Untuk fungsi, *identifier* menunjukkan apa yang dilakukan oleh fungsi tersebut.

6. Dalam dunia pemrograman, ada beberapa nama variabel yang menjadi kebiasaan untuk dipakai, misalnya nama variabel untuk mengunjungi elemen tabel, dipakai indeks i, j, dan k.
7. Konstanta dinamai dengan huruf kapital. Misalnya, PI.

3) Tipe Data

Komputer dapat mengolah data yang beragam. Pada dasarnya, data yang diolah oleh komputer, baik berupa numerik ataupun karakter, akan disimpan dalam bentuk biner. Oleh karena itu, nilai yang kalian masukkan dalam komputer pastilah akan disimpan dalam bentuk biner. Program perlu mengetahui bagaimana bilangan biner dibaca sehingga diperlukan suatu mekanisme untuk memberi tahu program tentang data yang kita simpan pada variabel tersebut. Hal ini diakomodir melalui tipe data. Suatu tipe data akan memiliki nama tipe, jenis data yang disimpan, dan rentang yang berbeda. Pada bahasa C, terdapat beberapa tipe data dasar yang dapat digunakan (Tabel 7.3).

Tabel 7.3 Beberapa Tipe Data Pada Bahasa C Beserta Ukuran Memori dan Rentang Nilainya

Nama Tipe	Jenis Data	Ukuran Memori	Rentang
int	Bilangan bulat	4 byte	-2.1×10^9 hingga 2.1×10^9
short	Bilangan bulat	2 byte	-32768 hingga 32767
long	Bilangan bulat	8 byte	-9.2×10^8 hingga 9.2×10^8
float	Bilangan riil	4 byte	1.2×10^{38} hingga 3.4×10^{38}
double	Bilangan riil	8 byte	2.3×10^{308} hingga 1.7×10^{308}
char	Karakter*	1 byte	-127 hingga 128

* Seperti yang termuat pada kode ASCII: ascii-code.com

Perhatikan bahwa rentang yang diberikan memungkinkan nilai negatif hingga positif, atau disebut tipe *data signed*. Apabila kalian menambahkan kata kunci *unsigned* di depan tipe data, tipe data tersebut hanya akan menampung bilangan positif dengan rentang dari 0 hingga $2^{\text{jumlah bit}} - 1$.

Praktik Baik Pemrograman

Gunakan tipe data yang sesuai dengan kebutuhan kalian. Sebagai contoh, saat mengolah data usia manusia dalam satuan tahun, kalian cukup menggunakan tipe *data short* yang memerlukan memori lebih kecil. Ketika kalian nanti membuat program yang mengolah dan menyimpan data dengan jumlah yang sangat besar, praktik ini dapat membuat program kalian berjalan dengan kebutuhan memori yang lebih efisien.

4) Variabel

Pada matematika, kalian mengenal variabel sebagai sebuah wadah untuk menyimpan suatu nilai. Variabel pada program memiliki fungsi yang sama. Nilai yang diberikan pada sebuah variabel akan disimpan di memori komputer. Komputer memberikan alamat pada lokasi memori tersebut yang sulit diingat oleh manusia. Oleh karena itu, variabel diberikan nama simbolik yang mudah untuk diingat oleh kalian dengan menggunakan *identifier.*

Dalam bahasa C, variabel perlu dideklarasikan dengan memberikan tipe data dan *identifiers* sebelum dapat digunakan. Deklarasi dapat dilakukan dengan menggunakan pernyataan berikut:

```
<tipe_data><nama_variabel>;
```

Pada saat deklarasi, variabel juga dapat diberikan nilai awal, misalnya dalam bentuk:

```
<tipe_data><nama_variabel> = <nilai awal>;
```

Variabel dengan tipe yang sama dapat dideklarasikan secara ringkas seperti:

```
<tipe_data><nama_variabel1>, <nama_variabel2>;
```

Beberapa contoh untuk mendeklarasikan variabel dapat dilihat pada Tabel 7.4.

Tabel 7.4 Beberapa Contoh Deklarasi Variabel

Tipe Data	Identifier	Deklarasi	Deklarasi dengan Nilai Awal
int	totalHarga	int totalHarga;	`int totalHarga = 150000;`
short	usia	short usia;	`short usia = 29;`
long	jumlahAtom	long jumlahAtom;	`long jumlahAtom = 9123151252214;`
float	jarak	float jarak;	`float jarak = 2.28;`
double	galat	double galat;	`double galat = 0.0000000001234;`
char	huruf	char huruf;	`` char huruf = `a`; ``

Deklarasi secara ringkas misalnya dapat dilakukan seperti berikut:

```
int panjang = 1, lebar = 2, luas;
  float alas, sisi, volume;
```

Tempat deklarasi variabel akan berpengaruh pada penggunaan variabel tersebut. Apabila deklarasi variabel dilakukan di dalam sebuah fungsi, variabel tersebut hanya dapat digunakan di dalam fungsi tersebut. Variabel seperti ini disebut variabel lokal. Apabila deklarasi dilakukan di luar fungsi, variabel tersebut akan dapat diakses di bagian program mana pun. Variabel ini disebut variabel global.

5) Konstanta

Berbeda dengan variabel yang nilainya dapat berubah, konstanta tidak dapat diubah. Saat dideklarasikan, nilai dari konstanta diberikan dan tidak dapat diubah kembali. Apabila kalian memaksa mengubah konstanta, kompilator akan memberikan pesan kepada kalian. Penggunaan konstanta yang lazim ialah untuk menyimpan nilai konstan seperti *pi* (π), *rho* (ρ), dan konstanta lainnya yang lazim digunakan.

Konstanta dapat dideklarasikan seperti variabel, dengan menambah kata kunci const di depan tipe data. Nilai awal harus langsung diberikan pada saat deklarasi. Misalnya, deklarasi konstanta pi dapat dilakukan sebagai berikut:

```
const float PI = 3.14;
```

6) Membaca dan Menulis

Untuk dapat membantu manusia, program harus dilengkapi dengan kemampuan berkomunikasi. Ada banyak cara untuk berkomunikasi lewat antarmuka pengguna (*user interface*), tetapi bentuk komunikasi dasar yang perlu kalian kuasai komunikasi lewat *command line interface* (CLI). Lewat CLI, kalian dapat berkomunikasi dengan sebuah program menggunakan teks, dan program pun akan merespons kalian dengan menggunakan teks. Dengan kata lain, interaksi menggunakan CLI sangat bergantung pada kemampuan program untuk membaca data yang diberikan oleh pengguna dan menuliskan hasil pekerjaan.

Agar dapat membaca dan menulis, program yang kalian buat perlu menggunakan *header* **stdio.h** yang memuat fungsi masukan-keluaran standar menggunakan CLI. Dua fungsi utama yang dapat digunakan ialah *scanf* untuk membaca dan `printf` untuk menulis. Untuk lebih jelasnya, perhatikanlah contoh program berikut. Pada program baca tulis berikut, kalian akan memerintahkan komputer untuk membaca suatu bilangan dan menuliskannya kembali.

```
/* Program Baca Tulis 3 */

#include<stdio.h>
int main(){
     int bilangan;
     scanf("%d", &bilangan);
     printf("Bilangan yang dibaca bernilai: ");
     printf("%d.\n", bilangan);
     return 0;
}
```

Pada baris keempat, program memanggil fungsi *scanf* untuk membaca masukan dari pengguna. Pada saat baris ini dieksekusi, program akan berhenti hingga pengguna memasukkan suatu bilangan dan menekan tombol *enter*.

Perhatikan bahwa pada baris tersebut, fungsi `scanf` menerima dua buah parameter, yaitu **%d** yang merupakan spesifikasi format (*format specifier*) dan &bilangan yang merupakan variabel untuk menampung nilai yang dibaca. Artinya, pada saat kalian menekan *enter*, program akan membaca nilai 10 yang kalian masukkan sebagai sebuah nilai bertipe data int, dan akan menyimpannya ke variabel bilangan. Di depan bilangan, *terdapat tanda ampersand (&) yang wajib digunakan untuk melakukan pembacaan*. Makna dari simbol & akan dijelaskan lebih detail pada kesempatan lain.

Setiap tipe data memiliki spesifikasi format yang dapat digunakan untuk menjelaskan jenis data kepada program. Ingat, program membaca dan menyimpan semua data sebagai bilangan biner. Spesifikasi format yang digunakan untuk tipe data pada bahasa C dapat dilihat pada Tabel 7.5.

Tabel 7.5 Spesifikasi Format pada Bahasa C

Nama Tipe	Spesifikasi Format (Signed)	Spesifikasi Format (*Unsigned*)
int	%d	%ud
short	%d	%ud
long	%ld	%uld
float	%d	%ud
double	%ld	%uld
char	%c	%uc

Baris kelima dan keenam adalah pernyataan untuk menulis menggunakan fungsi *printf*. Pada baris kelima, kalian memerintahkan program untuk mencetak suatu kalimat yang diapit dengan tanda petik ganda. Pada baris keenam, kalian memerintahkan program untuk mencetak nilai dari variabel bilangan. Mirip seperti fungsi `scanf`, kalian menemukan adanya spesifikasi format dan variabel. Bedanya, di depan variabel, tidak perlu ada tanda ampersand.

Saat dipanggil, program akan mencetak nilai dari variabel bilangan dengan format yang diberikan. Hanya saja, ada karakter baru yang muncul, yaitu **\n**. Ini adalah *escape sequence* yang digunakan untuk membuat garis baru, persis seperti ketika kalian menekan tombol *enter* pada aplikasi pengolah kata. Data yang ditulis setelah **\n** akan dicetak di baris yang baru oleh program.

Ada beberapa *escape sequence* lain yang dapat digunakan, yaitu:

```
\n (newline)
\t (tab)
\v (vertical tab)
```

```
\f (new page)
\b (backspace)
\r (carriage return)
\n (newline)
\a (beep, bell)
\\ (garis miring)
\" (tanda kutip ganda)
\` (tanda kutip)
```

Tentunya, kalian juga menggunakan fungsi `printf` untuk mencetak tipe data lain, misalnya bilangan riil. Pada bilangan riil, kalian dapat membatasi jumlah digit di belakang desimal dengan memodifikasi spesifikasi format. Perhatian kode berikut yang akan menampilkan angka 12.345 ke layar.

```
/* Program Cetak Desimal  */
#include <stdio.h>

int main() {
    float x=12.3456789;
    printf("%.3f\n", x);
    return 0;
```

Kalian juga dapat membaca atau menulis dua atau lebih nilai sekaligus seperti pada contoh berikut:

```
/* Program Cetak Bilangan Bulat dan Desimal */
#include <stdio.h>

int main() {
     int a, b;
     float c;

     scanf("%d %d %f", &a, &b, &c);
     printf("%d %d %.3f\n", a, b, c);
     return 0;
}
```

Ayo Cari Tahu

Ketikkan kode tersebut, berikan masukan 10 10 10.1234, dan perhatikan hasilnya.

Praktik Baik Pemrograman

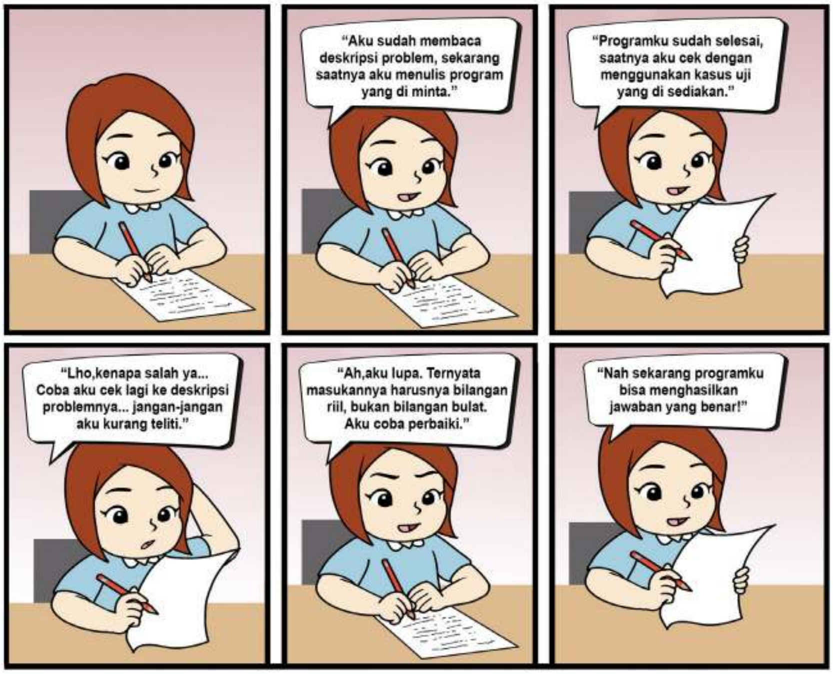
Seiring dengan makin kompleksnya program yang kalian buat, ada beberapa praktik baik yang dapat kalian lakukan.

1. Belajar menulis kode program memerlukan kemampuan bereksperimen dan mencoba hal baru. Karena keterbatasan halaman, buku ini tidak mungkin memberikan contoh selengkap mungkin untuk mencoba berbagai kemungkinan dalam menulis kode program. Oleh karena itu, kalian perlu melakukan eksperimen dan mencoba memodifikasi kode program yang telah diberikan. Kalian dapat membaca dokumentasi bahasa C yang dapat diakses di devdocs.io/c ketika menemukan suatu konsep ingin kalian pelajari lebih lanjut.
2. Jangan takut apabila program kalian gagal berjalan karena hal itu sangat wajar ketika belajar menulis kode program. Bahkan, pemrogram senior pun masih bisa melakukan kesalahan. Jadikan hal tersebut sebagai pemacu untuk mencari tahu hal yang membuat program kalian gagal berjalan. Dari situ, kalian akan menambah pengalaman dan di masa depan tidak akan kembali melakukan kesalahan yang sama.
3. Buat kode program kalian lebih mudah dibaca dengan menggunakan *whitespace* atau ruang kosong. *Whitespace* adalah istilah bagi karakter yang tidak tampak di layar (contoh: spasi, tab, dan newline). *Whitespace* dapat digunakan untuk membuat kode kalian lebih mudah dibaca dengan memberikan jarak pada kode. Jarak biasanya diberikan antara kumpulan baris kode yang memiliki peran berbeda. Contoh-contoh kode program yang diberikan pada unit ini disajikan dengan menggunakan whitespace yang dapat kalian adopsi.

Semangat terus untuk bereksperimen!

3. Belajar Menulis Program Sambil Menyelesaikan Masalah

Pada unit ini, aktivitas Ayo, Kita Berlatih diberikan sebagai bentuk latihan menyelesaikan suatu problem yang diberikan dengan pemrograman. Karena unit ini bersifat pengenalan pada kegiatan menulis kode program (coding), problem yang diberikan pada kalian diberikan dalam bentuk spesifikasi yang telah terstruktur. Pada kenyataannya, ketika kalian membuat program untuk menyelesaikan suatu permasalahan nyata, kalian perlu membuat sendiri spesifikasi dari program yang akan dibuat. Kalian perlu menetapkan sendiri tujuan dari program, format masukan, serta format keluaran dari program. Aktivitas ini akan kalian lakukan di akhir unit, tetapi sekarang kalian dapat fokus membuat program berdasarkan spesifikasi yang telah diberikan. Selain itu, pada unit pratik lintas bidang (PLB), kalian akan diajak untuk membuat sebuah program untuk menyelesaikan permasalahan di lingkungan tempat kalian tinggal.



(Sumber: Dokumen Kemendikbud,2021)

Pada setiap aktivitas, kalian akan diberikan setidaknya satu permasalahan untuk diselesaikan dengan menggunakan konsep yang telah diberikan pada buku. Bentuk permasalahan tersebut akan diberikan dengan struktur pada Tabel 7.6.

Tabel 7.6 Spesifikasi Program yang Diberikan pada Problem Latihan

Bagian	Penjelasan
Nomor dan Nama Problem	Identitas dari permasalahan.
Deskripsi Soal	Memberikan konteks permasalahan yang perlu diketahui oleh siswa dalam membuat program.
Format Masukan	Memberikan susunan data yang diberikan pada program oleh pengguna, beserta ukuran dari data tersebut.
Format Keluaran	Memberikan susunan informasi yang akan dikeluarkan oleh program kepada pengguna.
Contoh Masukan	Memberikan contoh data yang dimasukkan.
Contoh Keluaran	Memberikan contoh informasi yang dikeluarkan program berdasarkan data yang dimasukkan.

Bagian format keluaran dan format masukan menjadi penting karena auto-grader (jika kalian menggunakan ini di kelas) akan memeriksa kode program kalian dengan sangat ketat berdasarkan pasangan masukan-keluaran yang telah disiapkan, atau disebut kasus uji (test case). Apabila program kalian membaca

atau menulis dengan format yang salah, auto-grader tidak akan menganggap program kalian benar. Untuk membantu kalian memahami format tersebut, satu atau lebih contoh masukan dan keluaran akan diberikan.

Praktik Baik Pemrograman

Problem akan memberikan beberapa kasus uji. Akan tetapi, bukan berarti kalian hanya perlu mengecek program kalian dengan kasus uji yang ada di buku. Kalian perlu menguji program kalian dengan menggunakan berbagai kasus uji, yang kalian buat sendiri. Beberapa problem dirancang dengan adanya jebakan yang dapat membuat program kalian gagal berjalan pada kasus tertentu. Misalnya, ketika masukan yang diberikan memiliki rentang yang besar atau adanya kondisi yang dapat menyebabkan program berhenti.

Ayo, Kita Berlatih 3: Menulis dan Memperbaiki Program

Problem 1: Belajar Baca Tulis (Tingkat Kesulitan: ★★)

Deskripsi Soal:

Saatnya kalian berlatih untuk menulis kode program. Kalian akan membuat program untuk membaca tiga jenis bilangan dan mencetaknya kembali ke layar dengan format yang diberikan.

Format Masukan:

Satu baris yang berisi tiga buah bilangan yang dipisahkan oleh spasi. Bilangan pertama merupakan bilangan bulat positif dengan nilai maksimum 10,000. Bilangan kedua merupakan bilangan riil positif dengan nilai maksimum 10.0000. Bilangan ketiga merupakan bilangan bulat positif dengan nilai maksimum 100.

Format Keluaran:

Ada tiga baris. Baris pertama berisi bilangan pertama. Baris kedua berisi bilangan kedua dengan dua bilangan di belakang desimal. Baris ketiga berisi bilangan ketiga.

Contoh Kasus Uji

Masukan	Keluaran
10 20.1235 30	10 20.12 30

Problem 2: Bantulah Intan! (Tingkat Kesulitan: ★★★)

Intan ingin membuat program untuk mencetak kalimat berikut ke layer.

```
Andi berkata, "Satu, dua, tiga!".
```

Lalu, Andi pun menendang bola tersebut.

Akan tetapi, Intan tidak berhasil mencetak kalimat tersebut ke layar dengan persis sama. Berikut ini program yang ditulis oleh Intan.

```
#include <stdio.h>
int main(){
      printf("Andi berkata, "Satu, dua, tiga!".");
      printf("Lalu, Andi pun menendang bola tersebut.\n");
      return 0;
}
```

Perbaikilah program tersebut hingga berhasil mencetak kalimat yang benar!

Problem 3: Salah Baca (Tingkat Kesulitan: ★★)

Temukanlah kesalahan pada kode program berikut, kemudian perbaikilah kode berikut hingga dapat menghasilkan jawaban yang benar.

```
#include <stdio.h>

int main(){
      scanf("%c %c”, a, b);
      printf("Bilangan pertama: %c\d", a);
      printf("Bilangan kedua: %c\d", b);
      return 0;
}
```

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada bagian ini, kalian mendapatkan banyak konsep baru tentang program. Seperti apa perasaan kalian saat ini?
2. Apakah kalian bereksperimen dengan contoh-contoh yang diberikan di buku? Jika ya, pengetahuan paling menarik apa yang kalian temukan dari hasil eksperimen tersebut?
3. Pada saat memperbaiki program, apakah kalian dapat menemukan kesalahan pada program dengan mudah? Apakah kalian sebelumnya pernah melakukan kesalahan tersebut pada saat menulis program?
4. Bagaimana rencana kalian untuk menulis program setelah pertemuan ini? Apakah kalian sudah menentukan rencana berlatih secara mandiri agar dapat menulis program dengan lebih lancar dan terampil?

4. Ekspresi

Ekspresi merupakan bagian yang tidak terpisahkan dari program. Di dalam matematika, ekspresi terdiri atas kombinasi beberapa *operand* dan operator yang memiliki makna. Kalian telah terbiasa menulis ekspresi pada matematika,

misalnya dalam penjumlahan 10 + 5 yang melibatkan dua buah operand (10 dan 5) dan sebuah operator (+). Ekspresi pada pemrograman mirip dengan ekspresi yang kalian pelajari pada matematika, tetapi diperkaya dengan tambahan operator-operator untuk memudahkan kalian dalam menulis program.

Berdasarkan jumlah operand, suatu ekspresi dapat dibagi menjadi ekspresi:

1. *Unary* (satu *operand*), misalnya –a untuk menegasikan suatu bilangan.
2. *Binary* (dua *operand*), misalnya a+b untuk penjumlahan.
3. *Ternary* (tiga *operand*), misalnya a ? b : c yang akan dijelaskan lebih rinci.

Berdasarkan fungsinya, operator dalam bahasa C dapat dibagi menjadi operator penugasan, operator aritmatika, operator logika, serta *increment* dan *decrement*.

a. Operator Penugasan

Operator penugasan (simbol '=') untuk memberikan suatu nilai konstanta atau nilai yang diperoleh dari suatu ekspresi ke dalam variabel. *Operand* di sisi kiri akan menerima nilai dari *operand* di sisi kanan operator penugasan. Contoh penugasan telah kalian lakukan pada saat melakukan deklarasi variabel seperti:

```
int a = 5;
```

Perhatikan dan ingat baik-baik bahwa makna dari simbol '=' dalam bahasa C sangat berbeda dengan tanda "sama dengan" dalam matematika!

b. Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi matematika yang kalian kenal. Ada perbedaan notasi penulisan operator dengan yang kalian pelajari pada mata pelajaran Matematika yang dapat dilihat pada Tabel 7.7.

Tabel 7.7 Operator Matematika pada Bahasa C

Operasi Matematika	Contoh Ekspresi Aljabar	Operator Bahasa C	Ekspresi Bahasa C
Penjumlahan	1 + 2	+	1 + 2
Pengurangan	a - b	-	1 - b
Perkalian	ab	*	a * b
Pembagian	a/b	/	a / b
Modulo	a mod b	%	a % b

Penting: Hati-hati dalam melakukan pembagian pada bahasa C. Bahasa C sangat sensitif terhadap tipe data sehingga pembagian dua buah bilangan bulat akan menghasilkan bilangan bulat. Sebagai contoh, 10/3 pada kode program C saat dieksekusi akan menghasilkan nilai 3.

Seperti pada matematika, suatu ekspresi dapat terdiri atas banyak operator. Agar ekspresi tersebut dieksekusi dengan benar, diperlukan suatu urutan operasi (atau *operator precedence*). Urutan operasi dari operator pada bahasa C ialah:

1. Tanda kurung ()
2. Operator perkalian '*', pembagian '/', dan modulo '%'.

Selain operator-operator matematika tersebut, C juga memiliki operator-operator lain yang akan dipelajari di lain kesempatan. Daftar lengkap semua operator pada C dapat dilihat di http://en.cppreference.com/w/c/language/operator_ precedence

Hasil ekspresi dapat disimpan dalam suatu variabel menggunakan operator penugasan. Misalnya:

```
a = 5;
b = a + 1;
b = b + 5;
c = (a + b)*2 + a;
d = c % b;
```

Ekspresi seperti b = b + 5 mungkin tidak pernah kalian temui pada mata pelajaran Matematika. Akan tetapi, ekspresi ini sangat lazim ditemukan dalam menulis kode program. Makna dari ekspresi ini ialah kalian melakukan perubahan pada nilai suatu variabel berdasarkan nilai variabel pada saat ini. Hal ini telah kalian lakukan pada saat melakukan penelusuran pada Diagram Alir 4 di bagian awal unit ini.

Untuk ekspresi seperti ini, bahasa C menyediakan bentuk yang lebih ringkas yang disebut *compound assignment operator* atau operator penugasan majemuk. Misalnya, b = b + 5 dapat disingkat menjadi b += 5. Hal ini juga berlaku untuk operator matematika lainnya seperti b *= 2, c /= 3, dan seterusnya.

c. Operator Increment dan Decrement

Selain operator penugasan majemuk pada bagian sebelumnya, bahasa C juga memberikan operator *unary* yang lebih khusus untuk menambah (++) atau mengurangi nilai suatu variabel (--) dengan angka 1. Operator ini dapat diletakkan sebelum (*prefix*) atau setelah (*postfix*) operand.

Pada penulisan dalam bentuk prefix, perubahan nilai akan langsung dilakukan pada nilai variabel sebelum nilai variabel tersebut digunakan pada ekspresi. Sebagai contoh, setelah dua baris kode berikut dijalankan, nilai x akan bernilai 2 dan y akan bernilai 1.

```
int x = 1;
int y = ++x;
```

Sebaliknya, pada penulisan dalam bentuk *postfix*, nilai variabel akan digunakan terlebih dahulu pada ekspresi, baru perubahan dilakukan. Sebagai contoh, setelah dua baris kode berikut dijalankan, nilai x dan y akan bernilai 1.

```
int x = 1;
int y = x++;
```

d. Operator Logika, Relasional, dan Kesamaan

Di samping operator aritmatika, juga dikenal operator logika, relasional, dan kesamaan. Ekspresi yang menggunakan operator ini akan memiliki nilai benar (true atau dalam bahasa C bernilai tidak sama dengan 0) atau salah (false atau bernilai 0). Operator pada kategori ini memiliki peran yang sama dengan operator logika, relasional, dan kesamaan pada mata pelajaran Matematika. Operator tersebut pada bahasa C dapat dilihat pada Tabel 7.8.

Tabel 7.8 Operator Logika, Relasional, dan Kesamaan Pada Bahasa C

Aljabar	Bahasa C	Contoh	Makna
Operator Kesamaan / Pertidaksamaan			
=	==	a == b	Apakah nilai a sama dengan b?
≠	!=	a != b	Apakah nilai a tidak sama dengan b?
Operator Relasional			
>	>	a > b	Apakah a lebih besar dari b?
<	<	a < b	Apakah a lebih kecil dari b?
≥	>=	a >= b	Apakah a lebih besar atau sama dengan b?
≤	<=	a <= b	Apakah a lebih kecil atau sama dengan b?
Operator Logika			
AND	&&	a > 0 && b > 0	Apakah a dan b lebih besar daripada 0?
OR	||	a > 0 || b > 0	Apakah a atau b lebih besar daripada 0?
NOT	!	!(a > 0)	Apakah a tidak lebih besar daripada 0?

e. Operator Kondisional (*Ternary*)

Operator kondisional adalah operator ternary yang akan mengembalikan nilai berdasarkan suatu kondisi tertentu. Misalnya, kalian ingin mengembalikan nilai 1 jika suatu ekspresi a < 3 bernilai benar, dan mengembalikan nilai 0 jika sebaliknya. Kalian dapat menuliskannya dalam bentuk

```
n = ((a<3)?1:0)
```

Bentuk umum penggunaan operator ini adalah sebagai berikut:

```
(kondisi ? ekspresi-1 : ekspresi-2 )
```

Ekspresi 1 akan dikembalikan jika kondisi bernilai benar, sedangkan ekspresi 2 dikembalikan jika kondisi bernilai salah.

Praktik Baik Pemrograman

Pada saat menuliskan suatu ekspresi, gunakanlah pengelompokan dengan tanda kurung untuk membuat ekspresi tidak ambigu dan mudah untuk dibaca. Pastikan ekspresi tersebut ditulis dengan benar sebelum kalian menjalankan program. Biasakan untuk menguji (dalam pikiran kalian) sambil menulis, tidak menunggu program dijalankan.

Ayo, Kita Berlatih 4: Latihan Ekspresi

Problem 1: Menghitung Luas Tanah (Tingkat Kesulitan: ★★)

Deskripsi Soal:

Pak Algor memiliki sebidang tanah berbentuk segitiga siku-siku. Ia ingin mengetahui berapa luas tanah yang ia miliki. Bantulah Pak Algor menghitung dengan membuat program untuk menghitung luas tanahnya.

Format Masukan:

Dua buah bilangan bulat yang merupakan alas dan tinggi segitiga dari tanah Pak Algor.

Format Keluaran:

Sebuah bilangan riil, dengan dua digit di belakang desimal yang merupakan luas tanah Pak Algor. Akhiri keluaran dengan karakter newline.

Contoh Kasus Uji

Masukan	Keluaran
20 40	400.00

Problem 2: Menghitung Luas persegi (Tingkat Kesulitan: ★★)

Simak kembali Diagram Alir 1: Menghitung luas persegi pada bagian algoritma di unit ini. Buatlah program berdasarkan diagram alir tersebut. Panjang sisi yang diberikan ialah sebuah bilangan riil dengan nilai maksimum 1000 cm.

Problem 3: Hasil Bagi dan Sisa Pembagian (Tingkat Kesulitan: ★★★)

Deskripsi Soal:

Buatlah sebuah program untuk menampilkan hasil dan sisa pembagian dari dua buah bilangan bulat positif a dan b.

Format Masukan:

Dua buah bilangan positif a dan b yang dipisahkan oleh karakter spasi. Keduanya bernilai paling besar 10 miliar.

Format Keluaran:

Dua buah bilangan bulat yang ditulis di baris berbeda. Baris pertama adalah hasil pembagian, sedangkan baris kedua adalah sisa pembagian. Hasil pembagian dibulatkan ke bawah.

Contoh Kasus Uji

Masukan	Keluaran
1000 3	333 1

Problem 4: Benar atau Salah? (Tingkat Kesulitan: ★★★)

Ekspresi yang memuat operator logika, relasional, dan kesamaan dapat kalian telusuri tanpa menjalankan program. Berikut ini, diberikan beberapa ekspresi yang perlu kalian cek nilainya, jika diketahui nilai a = 1, b = 2, dan c = 3.

No	Ekspresi	Hasil
1	(a < b) || (b > c)	
2	(a >= b) || (b != c)	
3	(b == a) && (c > a)	
4	(b >= a) &&((b < c) || (c > a))	

Problem 5: Percantik Kode Program Ini! (Tingkat Kesulitan: ★★★★)

Uh... Kalian baru saja mendapatkan tugas untuk mempelajari sebuah kode program. Akan tetapi, kode program yang kalian terima tidak ditulis dengan menerapkan praktik baik pemrograman yang telah kalian pelajari sehingga sangat sulit untuk dipahami! Apalagi ternyata, saat kalian jalankan, program ini tidak bisa berjalan dengan benar.

```
#include <studio.h> int main(){ float jr2; float l; float
O; scanf("%f", jr2); l = 3.14*jr2*jr2; O = 2*3.14*jr2;
printf("%.2f %.2f\n",l, O); return 0; }
```

Sebagai seorang siswa teladan yang telah mempelajari praktik baik pemrograman, perbaikilah program di atas ke dalam bentuk yang menerapkan praktik baik pemrograman. Kemudian, jika ada, perbaikilah program tersebut sehingga dapat berjalan dengan benar. Setelah itu, apakah kalian dapat menebak apa yang dilakukan oleh program tersebut?

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada bagian ini, kalian mendapatkan banyak konsep baru tentang program. Seperti apa perasaan kalian saat ini?
2. Apakah kalian bereksperimen dengan contoh-contoh yang diberikan di buku? Jika ya, pengetahuan paling menarik apa yang kalian temukan dari hasil eksperimen tersebut?
3. Pada saat mengerjakan Problem 3, apakah kalian menyadari bahwa tipe data int tidak dapat menampung nilai hingga 10 miliar? Tipe data apa yang seharusnya digunakan?
4. Apa yang kalian rasakan saat memperbaiki program yang diberikan pada Problem 5? Apakah kalian makin menyadari pentingnya menerapkan praktik baik pemrograman?

5 Struktur Kontrol Keputusan

Pernahkah kalian ingin pergi ke sebuah tempat tertentu dengan menggunakan moda transportasi? Keputusan menggunakan sebuah moda transportasi untuk bepergian biasanya tergantung pada sebuah keadaan tertentu. Misalnya, apabila kondisi hujan, maka kalian akan lebih memilih menggunakan mobil daripada menggunakan sepeda motor, namun apabila cuaca sedang cerah dan jarak yang ditempuh adalah dekat, maka kalian akan memilih menggunakan sepeda motor.

Komputer merupakan alat yang membantu banyak aktivitas manusia. Pada dasarnya, komputer menjalankan perintah dari manusia. Perintah-perintah tersebut dituangkan secara tertulis dalam sebuah aturan tertentu yang disebut sebagai kode program yang bertujuan untuk mengatur bagaimana komputer harus bertindak untuk menyelesaikan sebuah permasalahan tertentu. Hal ini termasuk juga dalam proses pengambilan keputusan, seperti halnya dalam contoh pemilihan moda transportasi di atas.

Pada bagian ini kita akan mempelajari bagaimana pengambilan keputusan dilakukan dalam sebuah program. Istilah yang sering digunakan untuk ini adalah kondisional. Apa itu *kondisional*? Secara sederhana, kondisional adalah sebuah bentuk pernyataan "jika ..., maka ...". Pernyataan ini dibuat untuk mengekspresikan sebuah aksi berdasarkan sebuah kondisi tertentu. Sebagai contoh, ketika kita diminta untuk mengklasifikasikan sebuah bilangan merupakan bilangan ganjil atau genap, maka dapat kita membuat sebuah aturan sebagai berikut:

1. Jika bilangan tersebut habis dibagi 2, maka bilangan tersebut termasuk bilangan genap.
2. Jika bilangan tersebut tidak habis dibagi 2, maka bilangan tersebut termasuk bilangan ganjil.

Proses tersebut merupakan salah satu ilustrasi dari sebuah pernyataan kondisional. Pada bahasa pemrograman C, ada beberapa jenis pernyataan kondisional, misalnya pernyataan *if-else*, pernyataan *switch-case,* dan pernyataan yang bersarang.

a. Struktur Kontrol Keputusan *If - Else*

Ada beberapa variasi penggunaan struktur kontrol keputusan *If - Else*. Bentuk umum dari pernyataan if adalah sebagai berikut.

```
if (kondisi) {
    <pernyataan>;
    <pernyataan>;
    ....
}
```

Bagian kondisi dapat diisi dengan ekspresi yang menghasilkan nilai benar atau salah. Apabila kondisi menghasilkan nilai benar, semua pernyataan yang berada di dalam struktur kontrol keputusan tersebut akan dieksekusi oleh program. Sekarang, perhatikan program berikut, dan lakukanlah penelusuran untuk memeriksa keluaran dari program tersebut.

```
/* Program Membandingkan Bilangan */
#include <stdio.h>

int main(){
     int a = 1, b = 1;
     if (a == b) {
          printf("a sama dengan b \n");
     }
     return 0;
}
```

Program tersebut menggunakan struktur keputusan pada baris 6-8. Ekspresi yang digunakan pada bagian kondisi ialah a==b, sedangkan pernyataan yang dieksekusi jika kondisi benar terdapat pada baris 7.

```
/* Program Membandingkan Bilangan  */
#include <stdio.h>

int main(){
     int a = 1, b = 1;
     if (a == b)
          printf("a sama dengan b \n");
     return 0;
}
```

Struktur kontrol keputusan dapat menambahkan blok *else* yang akan dieksekusi apabila kondisi bernilai salah.

```
   if (kondisi)
<pernyataan>;
   else
<pernyataan>;
```

Misalnya, untuk mengecek apakah suatu bilangan merupakan bilangan ganjil atau genap, kalian dapat memanfaatkan struktur *if-else* sebagai berikut.

```
/* Program Cek Ganjil-Genap */
#include <stdio.h>
int main(){
      int bilangan;
      scanf("%d", &bilangan);
      if (bilangan % 2 == 0)
            printf("Bilangan Genap\n");
      else
            printf("Bilangan Ganjil\n");
      return 0;
}
```

Apabila kondisi makin kompleks, struktur *if-else* ini dapat dikembangkan kembali menjadi:

```
 if (kondisi ke-1)
     <statement>
 else if (kondisi ke-2)
     <statement>
    .....
 else if (kondisi ke-n )
     <statement>
 else
     <statement>
```

b. Struktur Kontrol Keputusan Switch-Case

Struktur kontrol keputusan yang memiliki cabang banyak dapat dibuat lebih sederhana menggunakan struktur *switch-case*. Bentuk umum dari struktur ini ialah sebagai berikut.

```
switch(switch_expr)
{
   case (constant expr1) :<statement>;
      <statement>;
      break;
   case (constant expr2) :<statement>;
      <statement>;
       break;
   .....
   default               :<statement>;
      <statement>;
      break;
}
```

Sebagai contoh, perhatikan kode program berikut:

```
#include <stdio.h>
int main() {
      int bilangan, sisaPembagian;
      scanf("%d", &bilangan);
      sisaPembagian = bilangan % 4;
      switch (sisaPembagian) {
            case 0: printf("Habis Dibagi\n"); break;
            case 1: printf("Sisa Satu\n"); break;
            case 2: printf("Sisa Dua\n"); break;
            case 3: printf("Sisa Tiga\n"); break;
      }
      return 0;
}
```

Pada program di atas, struktur *switch-case* memeriksa nilai yang ada pada variabel sisa pembagian. Karena nilai tersebut merupakan sisa pembagian sebuah bilangan dengan empat, hanya ada empat kemungkinan nilai, yaitu 0 sampai 3. Setiap kemungkinan nilai tersebut diperiksa melalui empat buah struktur case yang akan mencetak kalimat ke layar yang sesuai dengan sisa pembagian yang diperoleh.

c. Struktur Kontrol Keputusan Bersarang

Sebuah struktur kontrol dapat menjadi bagian dari suatu struktur kontrol lain. Hal ini disebut *nested* atau tersarang. Pada contoh berikut, diberikan sebuah kode program yang memiliki struktur kontrol keputusan bersarang. Telusurilah program tersebut jika program diberi masukan 1000 dan 10.

```
/* Program dengan IF tersarang */

#include <stdio.h>
int main() {
      int a, b;
      scanf("%d %d", &a, &b);
      if (b!=0)
            if (a/b>10)
                  printf("1\n");
      else
            printf("-1\n");
return 0;
}
```

Ayo, Kita Berlatih 5: Latihan Struktur Kontrol Keputusan

Problem 1: Membagi Bilangan (Tingkat Kesulitan: ★★)

Buatlah sebuah program dari Diagram Alir 3: Membagi bilangan yang tersedia pada bagian algoritma di awal unit ini.

Problem 2: Bilangan Bulat Positif (Tingkat Kesulitan: ★★)

Deskripsi Soal:

Buatlah program untuk mengecek apakah sebuah bilangan bulat adalah bilangan bulat positif.

1. Jika bilangan bulat tersebut merupakan bilangan bulat positif, maka cetaklah "Bilangan Bulat Positif".
2. Jika bilangan bulat tersebut bukan merupakan bilangan bulat positif, maka jangan cetak apa pun.

Format Masukan:

Sebuah bilangan bulat n. Nilai n berada pada rentang -100 < n < 100.

Format Keluaran:

Satu baris kalimat sesuai pada deskripsi soal

Contoh Kasus Uji

Masukan	Keluaran
3	Bilangan Bulat Positif
-10	

Problem 3: Jenis Bilangan Bulat (Tingkat Kesulitan: ★★★)

Deskripsi Soal:

Buatlah program untuk apakah sebuah bilangan bulat adalah bilangan termasuk bilangan bulat positif, negatif atau nol. Jika bilangan bulat tersebut merupakan bilangan bilangan bulat positif, cetak "Bilangan Bulat Positif". Jika bilangan bulat tersebut merupakan bilangan bulat negatif, cetak "Bilangan Bulat Negatif". Jika bilangan bulat tersebut merupakan bilangan bulat nol, cetak "Bilangan Bulat Nol".

Format Masukan:

Sebuah bilangan bulat n. Nilai n berada pada rentang -100 < n < 100.

Format Keluaran:

Satu baris kalimat sesuai pada deskripsi soal.

Contoh Kasus Uji

Masukan	Keluaran
3	Bilangan Bulat Positif
-5	Bilangan Bulat Negatif

Problem 4: Nama Bulan (Tingkat Kesulitan: ★★★)

Deskripsi Soal:

Buatlah sebuah program yang menerima masukan bilangan bulat yang berada pada rentang 1 - 12, dan akan mencetak nama bulan yang sesuai dengan bilangan bulat tersebut. Apabila bilangan berada di luar rentang tersebut, cetak kalimat "Tidak ada bulan yang sesuai".

Format Masukan:

Sebuah bilangan bulat n. Nilai n berada pada rentang -100 < n < 100.

Format Keluaran:

Satu baris kalimat sesuai pada deskripsi soal.

Contoh Kasus Uji

Masukan	Keluaran
1	Januari
3	Maret
13	Tidak ada bulan yang sesuai.

Problem 5: Mengecek Sisi Segitiga (Tingkat Kesulitan: ★★★★)

Tahukah kalian bahwa sebuah segitiga hanya bisa dibangun apabila sisi terpanjangnya lebih kecil daripada total panjang kedua sisi lainnya? Jika syarat ini tidak dipenuhi, tidak ada segitiga yang terbentuk.

Agria sedang membuat program untuk menghitung luas segitiga yang menerima masukan berupa tiga buah bilangan bulat yang merupakan panjang sisi segitiga tersebut. Akan tetapi, Agria menyadari bahwa ia harus terlebih dahulu memastikan ketiga panjang sisi yang dimasukkan benar-benar dapat membentuk sebuah segitiga. Oleh karena itu, ia merancang sebuah algoritma dalam bentuk diagram alir berikut untuk mengecek apakah ketiga bilangan tersebut dapat membentuk segitiga.

Tugas kalian adalah membantu Agria dengan membuat program berdasarkan diagram alir tersebut.

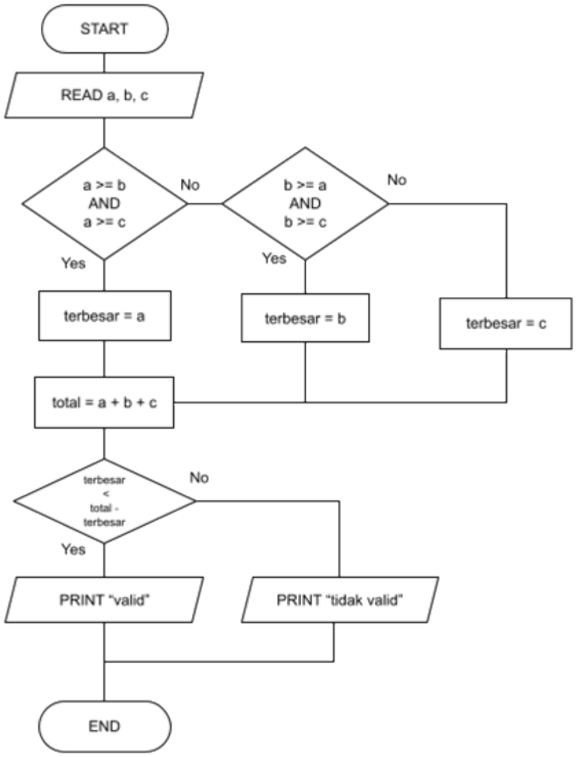
Problem 6: Belajar Membuat Kasus Uji (Tingkat Kesulitan: ★★★)

Perhatikan kembali diagram alir pada Problem 5. Diagram alir tersebut terlihat memiliki beberapa kemungkinan aliran, bergantung pada nilai masukan yang diberikan. Saat kalian mengecek program yang kalian buat dengan suatu kasus uji, kasus uji yang diberikan haruslah meliputi semua kemungkinan aliran tersebut. Sekarang, buatlah kasus uji sedemikian sehingga semua kemungkinan aliran pada diagram alir di atas dapat dicek.

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada bagian ini, kalian mendapatkan banyak konsep baru tentang program. Seperti apa perasaan kalian saat ini?
2. Apakah kalian bereksperimen dengan contoh-contoh yang diberikan di buku? Jika ya, pengetahuan paling menarik apa yang kalian temukan dari hasil eksperimen tersebut?
3. Dari latihan yang telah kalian kerjakan, terutama dari problem 1 dan 5, apa salah satu contoh penggunaan dari struktur kontrol keputusan?



4. Apakah kalian sudah memahami proses membuat kasus uji untuk menguji sebuah program, yang harus meliputi semua kemungkinan aliran pada diagram alir? Mengapa kalian harus memastikan semua kemungkinan aliran dicek?
5. Kesalahan apa yang sering kalian lakukan saat menulis kode program dengan menggunakan struktur kontrol keputusan?

6. Struktur Kontrol Perulangan

Kalian akan belajar memahami konsep dan cara kerja perulangan dalam pemrograman. Setelah menyelesaikan aktivitas ini, kalian diharapkan mampu menerapkan perulangan dan menghubungkannya dengan kejadian sehari-hari.

a. Pernahkah kalian melakukan aktivitas yang sama berulang kali? Eits... tidak perlu jauh-jauh, untuk melangkah saja, kita mengulang pergerakan kaki kanan dan kiri. Apakah kalian selalu berpikir sebelum melakukan setiap langkah? Pertanyaan terakhir tersebut dapat ditanyakan pada semua kegiatan perulangan yang kalian lakukan.
b. Membaca buku - kalian membalikkan halaman buku setiap selesai membaca halaman tersebut.

Salah satu keunggulan program komputer daripada manusia ialah kemampuannya untuk mengolah data yang berukuran besar atau melaksanakan suatu aksi berulang kali dalam periode waktu yang lama tanpa merasa bosan atau lelah. Hal ini dimungkinkan dengan adanya suatu kontrol perulangan. Pernyataan perulangan atau *loop* merupakan struktur program untuk keperluan iterasi, yaitu memproses satu atau beberapa pernyataan secara berulang (*looping*) berdasarkan kondisi tertentu. Program C menyediakan tiga bentuk pernyataan *loop,* yaitu:

1. for loop
2. while loop
3. do...while loop

a. Struktur Kontrol Perulangan for

Pernyataan ini umumnya digunakan untuk memproses pernyataan secara berulang-ulang, dengan jumlah perulangan yang dilakukan telah diketahui sebelumnya. Misalnya, berjalan sebanyak n langkah ke depan, atau mencetak barisan dari suku pertama hingga suku ke-n. Struktur kontrol perulangan for adalah sebagai berikut:

```
for (expr1; expr2; expr3)
{
   <statement>;
   ...
}
```

Struktur di atas akan dijalankan melalui proses berikut.

1. Ekspresi `expr1` akan dieksekusi ketika program menjalankan struktur *for* tersebut. Ekspresi ini biasanya berisi inisialisasi suatu variabel *counter* yang digunakan untuk menghitung jumlah perulangan yang telah dilakukan.
2. Ekspresi `expr2` merupakan suatu ekspresi bernilai benar atau salah (*boolean*) yang akan dicek sebelum pernyataan di dalam blok struktur dieksekusi. Apabila ekspresi ini bernilai benar, pernyataan akan dieksekusi. Sebaliknya, apabila ekspresi bernilai salah, pernyataan tidak akan dieksekusi dan perulangan berakhir. Dengan kata lain, pada bagian

ini, kalian menuliskan sebuah pernyataan yang merupakan kondisi berhenti (*stopping criteria*) untuk memastikan perulangan yang kalian buat memiliki langkah yang terbatas (dipastikan berhenti).

3. Ekspresi `expr3` merupakan sebuah pernyataan yang dijalankan setelah semua pernyataan di dalam struktur *for* dieksekusi. Biasanya, pernyataan ini dibuat untuk mengubah nilai variabel *counter* yang akan makin mendekati kondisi berhenti (memastikan nilai *counter* akan konvergen ke kondisi berhenti).

Walaupun ketiga ekspresi tersebut bersifat opsional (tidak harus ada), kalian disarankan untuk menuliskan ketiga ekspresi tersebut saat membuat program dengan jelas dan lengkap. Untuk saat ini, kalian perlu memahami teknik untuk menulis ketiga ekspresi tersebut dengan baik.

Perhatikan contoh kode program menulis bilangan bulat berikut. Pada contoh tersebut, program akan menulis bilangan bulat dari 0 hingga kurang dari n. Telusurilah kode program tersebut dengan teknik penelusuran yang telah kalian pelajari pada saat menelusuri suatu diagram alir!

```
/* Program Menulis Bilangan Bulat Sebanyak n Kali */

#include <stdio.h>
int main() {
      int i, n;
      scanf("%d", &n);
      for (i=0; i < n; i=i+1) {
          // i = i+1 dapat juga ditulis i++
          printf("%d\n", i);
      }
      return 0;
}
```

Keluaran program tersebut saat diberi masukan 5 adalah:

```
0
1
2
3
4
```

Pada contoh tersebut, variabel *counter* yang digunakan ialah i yang nilainya diinisialisasi (diberi nilai awal) dengan nilai 0. Pada bahasa C, lazimnya *counter,* dimulai dari nilai 0. Berbeda dengan proses pencacahan yang biasa kalian lakukan dari 1 di dunia nyata. Hal ini terkait dengan beberapa aspek teknis di bahasa C yang dimulai dari 0. Misalnya, di kelas XI nanti, kalian

akan menggunakan struktur data array yang dimulai dari indeks ke-0. Perlu diingat bahwa inisiasi dengan 0 ini hanyalah suatu kebiasaan masyarakat pemrogram dalam bahasa C, dan kalian tetap dapat melakukan pencacahan mulai dari 1. Kalian dapat mencoba mengubah kode program di atas sehingga *counter* berjalan dari 1. Selain *counter* yang berjalan menaik (*ascending*), kalian juga dapat membuat suatu *counter* yang berjalan turun (*descending*). Contoh ini disajikan misalnya pada Diagram Alir 4 dan 5 pada bagian algoritma.

Pernyataan yang ada pada expr1 hingga expr3 dapat ditulis menjadi deretan instruksi yang dipisahkan dengan tanda koma. Misalnya, terdapat pada contoh program berikut.

```
for (i=0, j=0; i<3; ++i, j++) {
      printf("%d %d", i, j);
}
```

Urutan pengerjaan akan sama seperti pada contoh sebelumnya. Akan tetapi, ada dua pernyataan yang akan dijalankan pada expr1 hingga expr3.

Praktik Baik Pemrograman

Pada saat merancang sebuah struktur perulangan for, kalian perlu memastikan agar kondisi berhenti pasti akan tercapai (konvergen). Apabila kondisi berhenti tidak pernah tercapai, akibatnya, struktur ini akan berjalan terus-menerus dan menyebabkan terjadinya suatu perulangan yang tidak terbatas (*infinite loop*). Apabila hal ini terjadi, program akan dibekukan oleh sistem operasi, bahkan akan dihentikan. Perhatikan contoh berikut.

```
/* Program dengan Perulangan tak Terbatas */

#include <stdio.h>

int main()
{
   int i, n;
   scanf("%d", &n);
   for(i=0; i<n; i--)
      printf("%d\n", n)
   return 0;
}
```

Dapat dilihat bahwa pada kode tersebut, nilai *counter* i akan berkurang dan tidak akan pernah melebihi nilai n jika n diisi dengan suatu bilangan bulat positif.

Ayo Cari Tahu

Lakukan penelusuran pada kode program tersebut dengan nilai n = 3.

b. Struktur Kontrol Perulangan While

Saat merancang perulangan, kalian bisa jadi tidak dapat menentukan berapa kali perulangan akan dilakukan. Akan tetapi, kalian mengetahui kondisi berhentinya. Misalkan instruksi berikut pada dunia nyata, “berjalan luruslah sampai ujung jalan, kemudian belok kiri.” Instruksi tersebut tidak memberikan gambaran jelas jumlah langkah yang akan kalian lakukan. Namun, secara naluriah, kalian mengetahui kapan kalian harus berhenti berjalan lurus, lalu berbelok ke arah kiri.

Pada program, suatu struktur kontrol *while* dikenal untuk melakukan perulangan seperti pada contoh di atas. Struktur kontrol tersebut dapat ditulis sebagai berikut. Pernyataan akan dieksekusi terus-menerus selama ekspresi kondisi bernilai benar.

```
while (ekspresi kondisi) {
     <pernyataan>;
   ...
}
```

Sebagai contoh, misalnya kalian akan menulis kode program untuk membaca dan menuliskan kembali bilangan bulat positif. Hal ini terus dilakukan hingga program membaca nilai -1. Program tersebut dapat kalian lihat di bawah ini.

```
/* Program Baca Tulis Bilangan   */

#include <stdio.h>
int main() {
      int bilangan;
      scanf("%d", &bilangan);
      while (bilangan != -1) {
            printf("%d\n", bilangan);
            scanf("%d", &bilangan);
      }
      return 0;
}
```

Ayo Cari Tahu

Lakukan penelusuran pada kode program ini dengan masukan yang diberikan ialah 1 2 3 4 -1.

c. Struktur Kontrol Perulangan Do – While

Struktur kontrol *do-while* memiliki perilaku yang mirip dengan while, yaitu kalian hanya mengetahui kondisi berhenti dari perulangan tersebut. Perbedaannya ialah struktur *do-while* dipastikan akan dikerjakan setidaknya satu kali. Bentuk umum pernyataan *do .. while* adalah sebagai berikut.

```
    do {
    <pernyataan>;
    } while (ekspresi kondisi);
```

Salah satu contoh penggunaan struktur *do-while* ialah ketika kalian menulis sebuah program interaktif yang akan meminta pengguna memasukkan kembali suatu nilai hingga nilai tersebut memenuhi suatu syarat. Hal ini akan sering kalian alami ketika kalian diminta untuk mengisi ulang (*retry*) saat menggunakan sebuah program atau mengisi sebuah formulir elektronik.

Misal, program berikut akan terus meminta pengguna memasukkan nilai sampai pengguna tersebut memasukkan bilangan bulat positif.

```
/* Program Verifikasi Masukan Pengguna */

#include <stdio.h>
int main() {
      int bilangan;
      do
            scanf("%d\n", &bilangan);
      while (!(bilangan > 0));
      printf("Anda  telah  memasukkan  bilangan  bulat
positif\n");
      return 0;
}
```

d. Struktur Kontrol Perulangan Bersarang

Sama seperti struktur kontrol keputusan, kalian dapat meletakkan struktur kontrol perulangan secara bersarang. Misalnya, pada contoh program berikut yang akan mencetak suatu pola berbentuk persegi menggunakan karakter asterisk '*'.

```
/* Program dengan FOR Bersarang  */

#include <stdio.h>
const int PANJANG = 2;
const int TINGGI = 3;
int main() {
      int i, j;
      for (i = 0; i < TINGGI; i++) {
            for (j = 0; j < PANJANG; j++) {
                  printf(“*”);
            }
            printf("\n");
      }
      return 0;
}
```

Ayo Cari Tahu

Lakukanlah penelusuran pada kode program tersebut!

Pada contoh di atas, *counter* pada struktur for terluar dan terdalam tidak saling berkaitan. Pada beberapa kasus nantinya, kedua *counter* tersebut bisa saja saling berkaitan. Misalnya, pada program berikut:

```
/* Program dengan Dua Counter */

#include <stdio.h>
int main() {
      int i, j, m, n;
      scanf("%d %d", &m, &n);
      for (i=1; i<=m; i++) {
            for (j=i; j<=n; j++) {
                  printf("%d", j);
                  if (j==n)
                        printf("\n");
                  else
                        printf(" ");
            }
      }
      return 0;
}
```

Pada program tersebut, kalian dapat melihat bahwa nilai awal *counter* j akan berpengaruh pada nilai *counter* i. Saat dijalankan, keluaran dari program tersebut ialah:

```
1 2 3 4 5
2 3 4 5
3 4 5
```

Tentunya, tidak ada batasan jumlah struktur yang kalian buat secara bersarang. Kalian pun juga dapat memadukan struktur perulangan dengan struktur keputusan sehingga menghasilkan program yang lebih kompleks.

Ayo, Kita Berlatih 6: Latihan Struktur Kontrol Perulangan

Problem 1: Menghitung Mundur (Tingkat Kesulitan: ★★)

Buatlah kode program berdasarkan Diagram Alir 4 pada bagian algoritma untuk mencetak bilangan secara hitung mundur.

Problem 2: Menghitung Rataan (Tingkat Kesulitan: ★★★)

Deskripsi Soal:

Buatlah sebuah program yang akan menghitung rata-rata dari n buah bilangan.

Format Masukan:

Baris pertama berisi sebuah bilangan bulat positif n yang menunjukkan banyaknya data, sedangkan baris berikutnya berisi n buah bilangan bulat. Nilai n maksimal 1000, dan besarnya bilangan yang harus dihitung rata-ratanya berada pada rentang -1 miliar hingga 1 miliar.

Format Keluaran:

Nilai rata-rata dari n buah bilangan masukan. Nilai rata-rata tersebut dituliskan sebagai bilangan riil dengan dua angka di belakang titik desimal.

Contoh Kasus Uji

Masukan	Keluaran
10 20 30	20.00

Problem 3: Mencari Bilangan Terbesar (Tingkat Kesulitan: ★★★)

Buatlah kode program berdasarkan Diagram Alir 5 pada bagian algoritma untuk mencari bilangan terbesar dari sekumpulan bilangan yang diberikan.

Problem 4: Membuat Mesin Sortir Kembang Kol (Tingkat Kesulitan: ★★★★)

Deskripsi Soal:

Kalian akan membantu seorang petani kembang kol untuk menyortir kembang kol yang telah dipanen berdasarkan ukurannya. Kembang kol tersebut akan dikelompokkan menjadi berukuran kecil (< 50 gram per buah), berukuran sedang (50-200 gram per buah), dan berukuran jumbo (> 200 gram per buah). Selama ini, petani tersebut menyortir kembang kol menggunakan tenaga manusia. Karena kalian telah memiliki kemampuan untuk membuat program

yang dapat melakukan hal tersebut secara otomatis, kalian pun mengajak petani tersebut untuk menerapkan *future practice* dengan menerapkan otomatisasi. Bantulah petani tersebut dengan program buatan kalian! Ini akan menjadi langkah awal bagi petani tersebut untuk menjadi seorang petani modern.

Format Masukan:

Program kalian akan membaca berat dari setiap kembang kol dalam satuan gram. Berat setiap kembang kol ditulis dalam bilangan riil. Apabila semua kembang kol telah tersortir, program kalian akan membaca nilai -1.

Format Keluaran:

Tiga buah bilangan bulat yang merupakan jumlah kembang kol pada setiap kategori, yaitu kecil, sedang, dan jumbo. Setiap bilangan ditulis pada baris yang berbeda.

Contoh Kasus Uji

Masukan	Keluaran
100.0 20.5 300.1 40.1 -1	2 1 1

Problem 5: Memperbaiki Program (Tingkat Kesulitan: ★★★★★)

Deskripsi Soal:

Buatlah sebuah program untuk menggambar sebuah pola X dengan menggunakan karakter asterisk (*).

Format Masukan:

Sebuah bilangan bulat n yang menyatakan ukuran pola yang akan dibuat. Nilai n berada pada rentang 1 hingga 20.

Format Keluaran:

Pola huruf X yang sesuai dengan ukuran yang dimasukkan. Perhatikan perbedaan untuk banyak baris ganjil dan genap pada contoh keluaran.

Contoh Kasus Uji

Masukan	Keluaran
5	*___* _*_*_ __*__ _*_*_ *___*
6	*____* _*__*_ __**__ _*__*_ *____*

Pada soal ini, kalian telah diberikan salah satu program yang dibuat untuk menyelesaikan permasalahan di atas. Akan tetapi, program tersebut tidak dapat dijalankan dan ditulis dengan tidak rapi! Pelajarilah kode program di bawah ini. Perbaiki program tersebut hingga dapat berjalan dengan benar. Sesuaikan pula penulisannya dengan mengikuti praktik baik yang telah kalian pelajari hingga saat ini.

```
int main(){
int N, i, j;
scanf("%d", &n);

for(i=0; i>n; i++){
for(j=0; j<n; j++){
if(j==i || j==n-i-1)printf("-");
else printf("*");
}
printf("\n");
}
```

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada bagian ini, kalian mendapatkan banyak konsep baru tentang program. Seperti apa perasaan kalian saat ini?
2. Apakah kalian bereksperimen dengan contoh-contoh yang diberikan di buku? Jika ya, pengetahuan paling menarik apa yang kalian temukan dari hasil eksperimen tersebut?
3. Pada Problem 6, apakah kalian memikirkan dampak sosial yang terjadi pada pekerja yang selama ini menyortir kembang kol secara manual tersebut? Jika kalian menjadi petani tersebut, apa yang akan kalian lakukan kepada pekerja tersebut? Apa yang kalian lakukan jika kalian menjadi pekerja tersebut?
4. Kesalahan apa yang sering kalian lakukan saat membuat program menggunakan struktur kontrol perulangan?

7. Fungsi

Selama ini, kalian telah sering bertemu dengan fungsi. Di Matematika, kalian membuat suatu persamaan menggunakan fungsi seperti $y = f(x) = x + 1$. Fungsi $f(x)$ menerima sebuah masukan x yang disebut fungsi daerah dan menghasilkan

nilai y yang merupakan fungsi wilayah. Fungsi sangat berhubungan dengan kemampuan abstraksi yang telah kalian pelajari di berpikir komputasional sehingga program yang kalian tulis dapat ditulis dengan lebih baik.

Sejauh ini, kalian juga telah menggunakan beberapa fungsi dalam berlatih pemrograman. Pada Diagram Alir, kalian telah mengenal sebuah simbol subprogram untuk memberikan abstraksi dari suatu proses lain yang kalian gunakan dalam solusi kalian. Pada program bahasa C, struktur main merupakan sebuah fungsi yang akan dieksekusi oleh sistem operasi ketika program dijalankan. Selain itu, kalian pun telah menggunakan fungsi seperti printf dan scanf.

Pada hakikatnya, fungsi pada program melambangkan suatu kumpulan pernyataan yang memiliki tujuan tertentu. Tujuan tersebut direpresentasikan oleh nama dari fungsi tersebut. Misalnya, scanf yang memiliki fungsi untuk membaca (*scan*) nilai dari pengguna. Fungsi juga dapat menerima parameter-parameter, dan juga dapat mengembalikan suatu nilai. Dengan membungkus kumpulan instruksi tadi ke dalam suatu fungsi, kalian dapat menggunakan kembali fungsi tersebut di berbagai lokasi dalam program kalian.

a. Membuat Fungsi

Saat membuat suatu fungsi baru, kalian perlu menentukan tiga hal tersebut: nama fungsi yang merepresentasikan tujuan dari fungsi, parameter yang dimasukkan ke dalam fungsi, serta nilai yang dikembalikan. Ketiga informasi ini disebut prototipe dari fungsi. Adapun pernyataan-pernyataan yang ada di dalam fungsi tersebut disebut implementasi dari fungsi.

Misal, kalian akan membuat sebuah fungsi untuk menghitung luas lingkaran. Maka, kalian perlu menetapkan ketiga hal tersebut dan menghasilkan prototipe fungsi berikut. Dari prototipe tersebut, kalian dapat melihat bahwa fungsi hitungLuasLingkaran memerlukan sebuah parameter dengan tipe *data float* yang merupakan radius dari lingkaran. Saat dipanggil, fungsi ini akan mengembalikan sebuah nilai float yang merupakan luas lingkaran.

```
float hitungLuasLingkaran(float radius);
```

Jika dituliskan lengkap dengan implementasinya, fungsi tersebut dapat ditulis menjadi:

```
float hitungLuasLingkaran(float radius) {
      float luas;
      luas = 3.14 * radius * radius;
      return luas;
}
```

Pada kode program tersebut, kata kunci return digunakan untuk mengakhiri fungsi dan mengembalikan suatu nilai. Dalam hal ini, nilai yang dikembalikan ialah luas lingkaran.

Perlu diingat bahwa prototipe dari fungsi-fungsi berikut akan dianggap sebagai fungsi yang berbeda karena memiliki parameter yang berbeda dari tipe data. Hal ini disebut *overloading*.

```
float hitungLuasLingkaran(float radius);
float hitungLuasLingkaran(int radius);
```

Kalian juga dapat membuat fungsi dengan jumlah parameter lebih dari satu. Misalnya, untuk menghitung luas persegi panjang berikut:

```
float hitungLuasPersegiPanjang(float panjang, float lebar);
```

Tentunya, kalian juga dapat membuat fungsi yang tidak memiliki parameter masukan, seperti yang kalian lakukan pada saat membuat fungsi main() pada program.

b. Memanggil Fungsi

Setelah dibuat, fungsi dapat dipanggil (function call) di dalam kode program. Perhatikan kode program berikut yang akan memanggil fungsi hitungLuasLingkaran yang telah dibuat.

```
/* Program Hitung Luas Bola  */

#include <stdio.h>
float hitungLuasLingkaran(float radius) {
      float luas;
      luas = 3.14 * radius * radius;
      return luas;
}

/* Program Hitung Luas Lingkaran */

#include <stdio.h>
float hitungLuasLingkaran(float radius) {
      float luas;
      luas = 3.14 * radius * radius;
      return luas;
}

int main() {
      float radius, luas;
      scanf("%f", &radius);
      luas = hitungLuasLingkaran(radius);
      printf("%f\n", luas);
}
```

Kalian juga dapat memanggil suatu fungsi di fungsi lain yang kalian buat. Misalnya, kalian ingin membuat fungsi untuk menghitung luas permukaan bola. Kalian dapat menulisnya menjadi:

```
float hitungLuasBola(float radius) {
      float luas;
      luas = 4.0 * hitungLuasLingkaran(radius);
      return luas;
}

int main() {
      float radius, luasBola;
      scanf("%f", &radius);
      luasBola = hitungLuasBola(radius);
      printf("%f\n", luasBola);
      return 0;
}
```

c. Variabel Lokal pada Fungsi

Pada contoh-contoh di atas, kalian akan menemukan adanya deklarasi variabel dengan *identifier* yang sama. Misalnya, variabel luas kalian temukan di dalam fungsi hitungLuasLingkaran dan hitungLuasBola. Kedua variabel tersebut disimpan pada alamat memori yang berbeda yang hanya bisa diakses di dalam fungsi tempat variabel tersebut berada. Dengan kata lain, keduanya ialah variabel lokal yang tidak saling berhubungan.

Praktik Baik Pemrograman

Gunakan fungsi untuk melakukan abstraksi. Kumpulkan fungsi-fungsi yang telah kalian buat agar dapat digunakan kembali untuk membuat program dengan lebih cepat. Kumpulan fungsi ini dapat kalian satukan menjadi sebuah pustaka atau *library*. Apabila pustaka tersebut memiliki manfaat yang besar dan dibutuhkan oleh banyak orang, kalian dapat membuat pustaka tersebut menjadi publik.

Ayo, Kita Berlatih 7: Latihan Fungsi

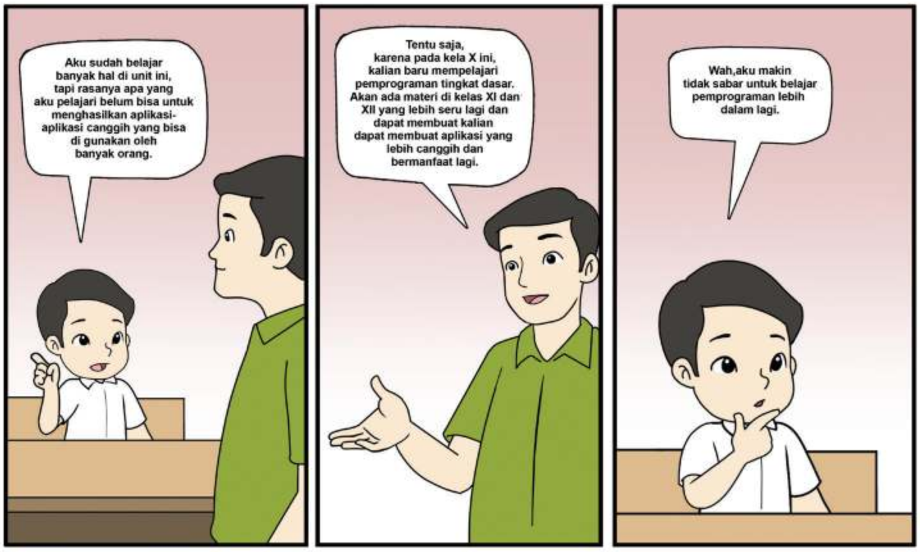
1. Buatlah kode program dari Diagram Alir 1 pada bagian algoritma, yaitu menghitung luas permukaan kubus.
2. Buatlah sebuah fungsi untuk menghitung luas dan keliling bangun datar, seperti persegi panjang, lingkaran, dan segitiga.
3. Buatlah sebuah fungsi untuk menghitung luas permukaan bangun ruang seperti balok, kerucut, bola, dan limas.

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada bagian ini, kalian mendapatkan banyak konsep baru tentang program. Seperti apa perasaan kalian saat ini?
2. Apakah kalian bereksperimen dengan contoh-contoh yang diberikan di buku? Jika ya, pengetahuan paling menarik apa yang kalian temukan dari hasil eksperimen tersebut?
3. Apakah kalian telah merasakan manfaat dari penggunaan fungsi?
4. Kesalahan apa yang sering kalian lakukan saat membuat fungsi?

8. Penutup: Coding dan Pemrograman



(Sumber: Dokumen Kemendikbud,2021)

Pada bagian-bagian sebelumnya, kalian telah melaksanakan beberapa kegiatan latihan untuk mempelajari cara menuliskan program menggunakan bahasa C. Apakah ini berarti kalian telah menguasai pemrograman? Belum tentu karena kompetensi utama yang kalian pelajari ialah kemampuan menulis kode program yang disebut *coding*.

Pada Tabel 7.8 berikut, kalian dapat melihat perbedaan antara coding dan pemrograman dari berbagai aspek. Dengan memperhatikan tabel tersebut, kalian akan menyadari bahwa kemampuan yang kalian latih selama ini ialah kemampuan *coding*. Kalian juga dapat melihat hal-hal yang harus kalian pelajari lebih lanjut untuk menguasai pemrograman. Walaupun demikian, karena unit ini disusun dengan adanya aktivitas menyelesaikan permasalahan, kalian telah menyentuh beberapa aspek pemrograman seperti merancang algoritma, menguji

kode program, dan memberikan solusi sederhana pada suatu permasalah yang diberikan. Perlu diingat juga, bahwa permasalahan diberikan pada kalian dalam bentuk spesifikasi program yang telah terstruktur. Pada pemrograman, kalian sendirilah yang harus merancang spesifikasi program tersebut.

Tabel 7.8 Perbedaan antara Coding dan Pemrograman

Aspek	Coding	Pemrograman
Definisi	Kegiatan menulis kode program dengan menggunakan bahasa pemrograman tertentu.	Proses mengembangkan sebuah solusi program untuk menyelesaikan masalah tertentu.
Perangkat yang Digunakan	Editor teks sudah cukup.	Perangkat analisis, desain, editor, dan pengujian.
Keahlian yang Dibutuhkan	Kemampuan membaca dan mengetik (menuliskan) sintaks bahasa pemrograman dengan benar.	Kemampuan berpikir komputasional, merancang algoritma, memodelkan permasalahan, dan manajemen proyek.
Hasil	Kode sederhana dengan banyak batasan.	Program yang siap digunakan oleh pengguna.

Pada unit PLB, kalian akan merasakan kegiatan pemrograman langsung untuk menyelesaikan permasalahan di sekitar kalian. Untuk mempersiapkan kalian mengikuti kegiatan PLB tersebut, di akhir unit ini, ada beberapa permasalahan yang diberikan untuk kalian kerjakan. Spesifikasi program tidak diberikan sehingga kalian perlu memanfaatkan kemampuan berpikir komputasional kalian dengan lebih maksimal.

Ayo, Kita Berlatih 8: Latihan Pemrograman

Pada bagian ini, kalian akan diberikan beberapa problem yang harus kalian selesaikan melalui pemrograman. Gunakan kemampuan berpikir komputasional kalian untuk menganalisis permasalahan yang diberikan. Rancanglah strategi dalam bentuk algoritma untuk menyelesaikan permasalahan tersebut. Kemudian, implementasikan algoritma tersebut ke dalam bentuk program menggunakan bahasa C.

Problem 1. Mencetak Faktor Prima (Tingkat Kesulitan: ★★★★)

Kalian pasti telah mengenal faktor prima dari suatu bilangan. Misalnya, 20 akan memiliki faktor prima 2 dan 5. Rancang dan buatlah sebuah program untuk mencetak faktor prima dari sebuah bilangan bulat yang diberikan.

Problem 2. Mengecek Bilangan Prima (Tingkat Kesulitan: ★★★★)

Apakah 713 merupakan bilangan prima? Rancang dan buatlah sebuah program yang dapat kalian gunakan untuk mengecek keprimaan suatu bilangan bulat yang diberikan.

Problem 3. Mengecek Tanggal (Tingkat Kesulitan: ★★★★)

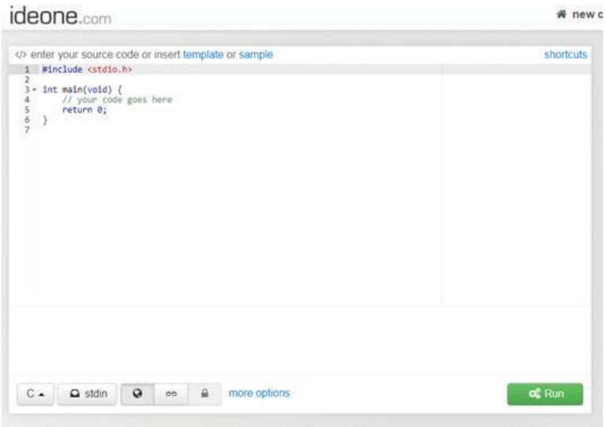
Menulis tanggal tidak boleh sembarang. Ada aturan-aturan yang berlaku. Misalnya, ada aturan tahun kabisat serta jumlah hari dalam suatu bulan yang telah ditentukan. Pada problem ini, tugas kalian ialah merancang dan membuat sebuah program yang dapat memeriksa apakah sebuah tanggal dengan format yang kalian rancang, saat diberikan oleh pengguna dinyatakan benar.

D. Pengayaan

1. Menggunakan IDE Daring

Pada aktivitas ini, siswa dapat menggunakan kompilator atau *compiler* untuk membuat program. *Compiler* berfungsi untuk membaca dan menerjemahkan program yang kita tulis agar dapat dimengerti oleh komputer. Salah satu *online compiler* yang dapat siswa gunakan ialah *ideone* dengan cara mengunjungi situs https://ideone.com/ . Berikut ini tampilan yang akan kalian lihat.

Skenario



Gambar 7.5 Tampilan *Ideone*

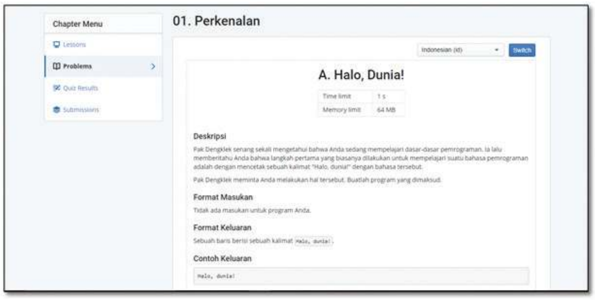
Sumber: Kemendikbud, 2021

Pada bagian bawah, terdapat beberapa menu yang dapat kita gunakan. Tombol paling ujung kiri berfungsi untuk memilih bahasa pemrograman yang akan digunakan. Terdapat berbagai macam bahasa pemrograman yang tersedia, tetapi yang akan digunakan di sini ialah bahasa C. Jadi, pastikan tombol paling ujung kiri bertuliskan "C". Kemudian, di sebelahnya, terdapat tombol *stdin*. Tombol ini berfungsi untuk memasukkan *input* yang akan kita gunakan pada Aktivitas 3. Tombol berwarna hijau di ujung kanan berfungsi untuk menjalankan atau mengeksekusi program yang kita buat.

2. Berlatih Pemrograman Secara Mandiri Menggunakan Auto-Grader

Dalam dunia pemrograman, terutama dalam konteks berlatih atau berkompetisi, dikenal sebuah program yang dapat digunakan untuk mengecek apakah program yang kalian buat dapat menyelesaikan permasalahan yang diberikan. Program tersebut disebut *auto-grader* (penilai otomatis). Program tersebut lazim dimiliki oleh berbagai web tempat berlatih pemrograman untuk memudahkan murid memastikan program yang ia buat benar. Terdapat beberapa web yang dapat digunakan untuk berlatih pemrograman secara mandiri, salah satunya dikembangkan oleh Tim Olimpiade Komputer Indonesia yang dapat diakses di tlx.toki.or.id.

Web-web tersebut biasanya terdiri atas kumpulan materi dan soal yang dapat diselesaikan oleh kalian. Suatu soal, atau biasa disebut problem, terdiri atas deskripsi soal, format masukan, format keluaran, penjelasan, serta batasan yang digunakan dalam permasalahan tersebut. Setelah menganalisis dan memahami soal tersebut, kalian dapat merancang suatu algoritma untuk menyelesaikan permasalahan tersebut dalam batasan yang diberikan. Setelah itu, kalian dapat membuat kode program yang dapat digunakan untuk menyelesaikan soal tersebut dan mengirimkannya ke *auto-grader*.



Gambar 7.6 Contoh Soal yang Terdapat pada Web TLX yang Dikelola oleh Ikatan Alumni Tim Olimpiade Komputer Indonesia (IA-TOKI)

Sumber: Dokumen Kemendikbud, 2021

Auto-grader akan menjalankan program yang telah dibuat, kemudian memberikan suatu kumpulan masukan pada program tersebut. Hasil dari program dibandingkan dengan hasil yang benar, yang telah disimpan pada *auto-grader* oleh pembuat soal. Pasangan masukan-keluaran yang dimasukkan oleh pembuat soal ini disebut kasus uji (*test case*). Suatu soal bisa memiliki satu hingga banyak *test case* bergantung pada kompleksitas dan tingkat kesulitan soal tersebut.

Berdasarkan hasil pengujian tersebut, *auto-grader* akan memberikan skor pada program kalian. Skor diberikan berdasarkan persentase *test case* yang dijawab dengan benar oleh program yang kalian buat. Apabila program kalian menghasilkan lima jawaban benar dari sepuluh *test case* yang tersedia, kalian akan mendapatkan skor sebesar 50. Apabila semua *test case* dijawab dengan benar, kalian mendapatkan nilai 100 dan mendapatkan hasil ACCEPTED.

Selain hasil ACCEPTED, terdapat beberapa kemungkinan lain yang dikeluarkan oleh *auto-grader* pada program kalian. Misalnya, WRONG ANSWER diberikan apabila program kalian tidak mendapatkan nilai 100. TIME LIMIT EXCEEDED diberikan apabila program kalian berjalan lebih lama daripada yang diminta oleh pembuat soal. MEMORY LIMIT EXCEEDED diberikan apabila program kalian menggunakan memori yang lebih besar daripada yang diberikan pada soal. RUN TIME ERROR apabila program kalian berhenti secara tidak wajar saat dijalankan. COMPILE ERROR apabila kode program kalian gagal dikompilasi oleh *auto-grader*. Selain itu, masih terdapat hasil-hasil lainnya yang bergantung pada masalah yang terjadi pada saat kode program dijalankan.

Kalian dapat berlatih soal-soal yang tersedia pada kursus pemrograman dasar yang tersedia di TLX (tlx.toki.id/courses/basic). Kalian dapat membuat akun dan mulai mengikuti kursus tersebut dengan membaca materi dan mengerjakan soal-soal yang diberikan. Selain berupa modul, terkadang IA-TOKI juga mengadakan kontes pemrograman dengan berbagai tingkat kesulitan yang dapat kalian ikuti.

Setelah menamatkan materi pemrograman dasar, kalian dapat melanjutkan ke latihan-latihan yang lebih kompleks, yang dirancang untuk kompetisi-kompetisi pemrograman (tlx.toki.id/courses/competitive). Di tingkat SMA, ada International Olympiad Informatics (IOI), sedangkan di tingkat perguruan tinggi, terdapat International Collegiate Programming Contest (ICPC). Kedua kompetisi ini adalah sebuah gold-standard dari kompetisi pemrograman tingkat internasional.

Pemrograman merupakan kemampuan yang harus terus dilatih, dan tidak cukup dengan dihafal. Makin sering kita berlatih, dan semakin banyak soal yang kita kerjakan, kita akan makin cepat dan mudah mengerjakan soal dengan tingkat kesulitan yang lebih tinggi. Apabila kalian menemukan kesulitan, terdapat banyak forum serta media *online* dimana kalian dapat mencari petunjuk untuk membantu kalian.

3. Perbandingan Sintaks Bahasa C dan Python

Ada banyak bahasa pemrograman dengan kelebihan dan keunggulan masing-masing. Bahasa C pada unit ini hanyalah salah satu bahasa dari banyak bahasa pemrograman yang dapat dipilih. Pada tingkat SMP, serta pada unit analisis data di SMA Kelas X, diperkenalkan juga bahasa pemrograman lain, yaitu Python. Walaupun secara sintaks berbeda, struktur kontrol kedua bahasa tersebut mirip. Untuk mempermudah, diberikan Tabel 7.9 berikut yang memberikan sintaks pada bahasa C dan Python untuk mengerjakan suatu hal yang sama. Kalian dapat mempelajari lebih lanjut mengenai bahasa Python di python.org.

Tabel 7.9 Padanan Sintaks Bahasa C dan Bahasa Python

<table>
<tr><th>Sintaks pada Bahasa C</th><th>Sintaks pada Bahasa Python</th><th>Penjelasan</th></tr>
<tr><td>printf("%d", x);</td><td>print(x)</td><td>Mencetak keluaran</td></tr>
<tr><td>scanf("%d", &x);</td><td>x = input()</td><td>Membaca masukan</td></tr>
<tr><td>x = 10;</td><td>x = 10</td><td>Pemberian nilai</td></tr>
<tr><td>a = (x + y)*z;</td><td>a = (x + y)*z</td><td>Operasi aritmetika</td></tr>
<tr><td>< <= > >= == !=</td><td>< <= > >= == !=</td><td>Operasi perbandingan</td></tr>
<tr><td>&&, ||</td><td>and , not</td><td></td></tr>
<tr><td>if (<kondisi>)
{
pernyataan;
}</td><td>if <kondisi>:
    pernyataan</td><td>Percabangan dengan satu pernyataan</td></tr>
<tr><td>if (<kondisi1>)
{
pernyataan1;
}
else if (<kondisi2>)
{
pernyataan2;
}
else
{
pernyataan3;
}</td><td>if <kondisi1>:
    pernyataan1
elif <kondisi2>:
    pernyataan2
else:
    pernyataan3</td><td>Percabangan dengan lebih dari satu kemungkinan pernyataan</td></tr>
<tr><td>while (<kondisi>)
{
pernyataan;
}</td><td>while <kondisi>:
    pernyataan</td><td>Perulangan dengan while (berdasarkan kondisi)</td></tr>
<tr><td>for (int i=0;i<n;i++)
{
  pernyataan dengan i;
}</td><td>for i in range(n):
    pernyataan dengan i</td><td>Perulangan dengan iterator (variabel penunjuk)</td></tr>
</table>

Contoh Program: Menghitung Jumlah Bilangan dari 1 hingga N

Program dalam bahasa C:

```
/* Program Hitung Jumlah Bilangan 1 s/d N dalam Bahasa
C */

#include <stdio.h>
int main() {
      int n, jumlah;
      scanf("%d", &n);
      jumlah = 0;
      for (int i = 0; i < n; i++)
            jumlah = jumlah + i;
      printf("jumlah bilangan 1 s/d %d = %d\n", n - 1,
jumlah);
      return 0;
}
```

Program dalam bahasa Python:

```
# Program Hitung Jumlah Bilangan 1 – N di Python

n = input()
jumlah = 0;
for i in range(n):
      jumlah = jumlah + i
print("jumlah bilangan 1 s/d %d = %d\n" %( n - 1,
jumlah))
```

Menerjemahkan sebuah program dari satu bahasa ke bahasa lain itu mudah, bukan? Semudah kalian menerjemahkan “Saya cinta Indonesia” menjadi “I love Indonesia”!

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021

Informatika untuk SMA Kelas X

Penulis : Auzi Asfarian, Mushthofa

ISBN : 978-602-244-506-7

Bab 8
Dampak Sosial Informatika



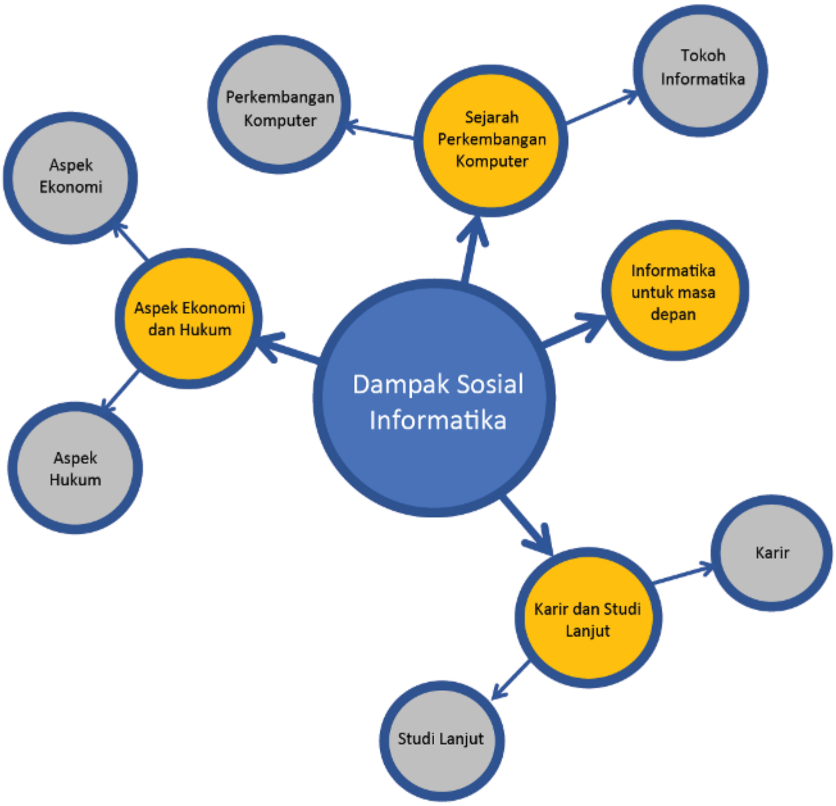
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu memahami sejarah perkembangan komputer dan tokoh-tokohnya, aspek ekonomi dan hukum produk Informatika, merancang gagasan solusi informatika, dan memahami studi lanjut dan karier di bidang informatika.

Pertanyaan Pemantik

Apa cita-cita kalian yang berkaitan dengan informatika? Bagaimana kalian akan mencapainya?

Peta Konsep



Gambar 8.1 Peta Konsep Dampak Sosial Informatika
Sumber: Dokumen Kemendikbud, 2021

Apersepsi

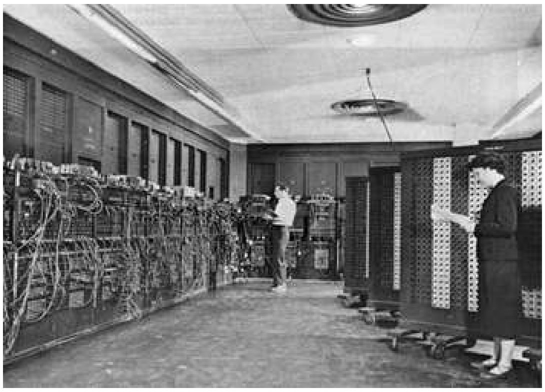
Saat ini, informatika sangat memengaruhi kehidupan manusia. Informatika tumbuh dari sejarah perkembangan komputer di masa lampau dengan tokoh-tokoh jeniusnya. Informatika menghasilkan produk yang bisa berdampak pada aspek ekonomi maupun hukum. Di masa depan, kalian tidak boleh tertinggal untuk tahu teknologi di bidang Informatika ini, untuk membantu menyelesaikan permasalahan yang penting.

Kata Kunci

sejarah informatika, aspek ekonomi, aspek hukum, studi lanjut, sertifikasi, karier

A. Sejarah Perkembangan Komputer

Komputer pada saat ini telah masuk pada seluruh sendi kehidupan kalian. Komputer saat ini sangat terjangkau dan memiliki banyak bentuk. Komputer tidak lagi berbentuk *desktop*, tetapi juga dapat berbentuk laptop, ponsel pintar, jam tangan, kacamata, bahkan menyatu dengan beragam benda yang kita sebut sebagai *internet of things*. Akan tetapi, tidak banyak yang mengetahui bahwa komputer memiliki sejarah yang sangat panjang. Ternyata, sejarah mesin komputasi dimulai bahkan jauh sebelum Masehi, yaitu +/- 2400 BC dengan abacus, sebelum mesin elektronik pertama diciptakan. Komputer kemudian dapat diprogram, yang dirancang oleh Charles Babbage, dengan pemrogram pertamanya ialah seorang wanita bernama Ada Lovelace.



Gambar 8.2 Electronic Numerical Integrator and Computer (ENIAC), komputer elektronik pertama yang dapat diprogram

Model komputasi yang melahirkan komputer elektronik yang dipakai saat ini ialah *Mesin Turing*. Mesin ini diciptakan oleh Alan Turing. Mesin Turing merupakan model komputasi, sebuah mesin abstrak yang mengolah simbol pada sepotong pita dengan mengacu ke sebuah tabel aturan. Walaupun sangat sederhana, diberikan sebuah algoritma komputer, sebuah Mesin Turing dapat menyimulasi logika algoritma yang dirancang. Mesin abstrak sederhana ini menjadi dasar pengembangan bahasa-bahasa pemrograman dan teori otomata yang mendasari otomasi. Perhatikan bahwa perkembangan perangkat keras dan perangkat lunak selalu dimulai dari model abstrak. Ini membuktikan, bahwa berpikir komputasional selalu mendahului dan mendorong lahirnya teknologi.

Sejak itu, beragam benda untuk membantu perhitungan ditemukan, dengan kompleksitas dan kinerja yang makin tinggi. Perubahan komputer yang berlangsung terus sepanjang waktu dan secara berjenjang dipacu dengan perkembangan ilmu pengetahuan membuat komputer berevolusi.

Di tingkat SMP, siswa telah mempelajari sistem komputer, dan mengenal bahwa komputer bisa terdiri atas komputer personal, mini personal, mini komputer dan *mainframe*. Siswa juga sudah mempelajari berbagai jenis *software*, mulai sistem perangkat lunak, *development tools* (compiler, IDE-Integrated Development Environment seperti yang dipakai untuk pemrograman

dengan scratch dll), dan aplikasi (*email*, pemroses kata, pemroses lembar kerja, presentasi, *browser*, laboratorium virtual dan lain-lain). Siswa sudah mengalami bagaimana mudahnya menggunakan sistem komputer di zaman sekarang, yang sangat berbeda dengan komputer saat pertama kali ditemukan. Ini sama dengan perkembangan alat transportasi, misalnya dimulai dari kereta kuda, mobil kuno sampai mobil modern tanpa awak sekarang.

Mempelajari sejarah sangat penting untuk belajar dari masa lalu. Belajar tentang keberhasilan dan kegagalan, serta betapa mesin dan teknologi yang ditemukan akan mempercepat penemuan mesin-mesin baru dan komponen-komponen yang lebih tangguh, dengan ukuran yang makin kecil, tetapi kapasitas dan kinerjanya makin tinggi. Mesin komputer juga lahir dari konsep.



Gambar 8.3 Ada Lovelace, Pemrogram Komputer Paling Awal di Dunia.

Perkembangan teknologi juga tak lepas dari perkembangan ilmuwan dan insinyur yang mengembangkannya. Tokoh-tokoh tersebut tidak hanya menciptakan mesin, tetapi juga menciptakan konsep-konsep yang dipakai sampai sekarang, misalnya Algoritma ditemukan pada tahun 830 oleh Al-Khawarizmi.

Di tingkat SMP, kalian juga sudah dilatih untuk merangkum bahan bacaan digital. Kali ini, kalian diminta untuk merangkum informasi tentang perkembangan komputer (*hardware* dan *software*), dan tokoh-tokoh yang berperan di balik perkembangan tersebut. Karena banyak, kalian harus memilih suatu periode tertentu yang "pendek" dan paling menakjubkan buat kalian, dan mengulas mesin yang ada saat itu berikut tokoh-tokohnya.



Gambar 8.4 Beragam bentuk komputer dalam kehidupan sehari-hari (a) Smart Watch (b) Surface Table, (c) Virtual Reality.

Ayo Berdiskusi

Menyelami Sejarah Komputer

Pada aktivitas ini, kalian diminta untuk menelaah kejadian, tokoh, atau benda yang menjadi bagian penting sejarah perkembangan komputer. Berdasarkan hasil penelaahan, buatlah suatu paparan kreatif dengan menggunakan beragam peraga untuk menyampaikan hasil penelaahan di depan kelas. Ikutilah petunjuk dari guru.

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Hal baru apa mengenai sejarah komputer yang kalian dapatkan hari ini?
2. Apakah kalian sudah memahami bahwa komputer merupakan alat untuk membantu manusia menjalankan gagasan hasil berpikir komputasionalnya?
3. Menurut kalian, apa dampak komputer yang paling penting bagi perkembangan peradaban manusia?
4. Menurut kalian, peristiwa, tokoh, atau benda apa pada sejarah komputer yang paling menarik? Mengapa?

B. Aspek Ekonomi dan Hukum dari Produk Informatika

Sebuah produk informatika memiliki aspek ekonomi dan dampak hukum. Bagaimana aspek ekonomi dan hukum dari sebuah produk informatika? Mari, kita cermati berikut ini.

1. Aspek Ekonomi

Industri perangkat keras dan perangkat lunak berkembang sejak ditemukannya komputer. Mula-mula, perusahaan perangkat keras *mainframe* dan mini-komputer yang muncul, misalnya IBM, Hitachi, UNIVAC, dan berbagai merk lainnya. Kemudian, dengan lahirnya PC (*personal computer*) dan diikuti dengan *laptop*, *tablet* serta telepon pintar, industri perangkat keras mulai tumbuh. Bukan hanya komputer, tetapi juga semua *periferal* seperti *printer*, alat *input/output* lainnya. Industri perangkat keras melahirkan industri manufaktur perangkat keras. Sayangnya, di Indonesia masih dalam tahap perakitan.

Seiring dengan berkembangnya industri di bidang perangkat keras, industri di bidang perangkat lunak juga berkembang. Mula-mula, industri perangkat lunak menyatu dengan perangkat keras, dan perangkat lunak dibangun untuk perusahaan atau pemerintahan, bukan untuk kehidupan

individu sehari-hari. Ini juga diwarnai dengan cara kerja manusia dengan komputer *mainframe* dan mini-komputer, di mana pengguna tidak berhubungan langsung dengan perangkat keras. Generasi selanjutnya ialah manusia mulai dapat berinteraksi dengan komputer lewat kabel atau jaringan lokal dengan menggunakan terminal berbasis tabung. Komputer hanya dipakai para profesional di kantor, atau untuk para periset di laboratorium. Dengan lahirnya PC, komputer makin dipakai di rumah dan untuk kehidupan sehari-hari, perkembangannya didukung dengan lahirnya internet. Komputer sekarang sudah dalam genggaman manusia dalam bentuk telepon pintar. Industri perangkat lunak berkembang pesat untuk memenuhi kebutuhan manusia berinteraksi dengan komputer, atau manusia berinteraksi dengan manusia lainnya baik dalam kelompok maupun individual.

Berbeda dengan industri manufaktur perangkat keras, makin banyak diciptakan perangkat lunak. Apalagi dengan makin murahnya komputer dan perangkat lain sampai dengan telepon pintar, perangkat lunak makin banyak menunjang kehidupan sehari-hari dalam berbagai bidang. Perangkat lunak diperlukan menunjang kehidupan sehari-hari misalnya untuk berkomunikasi dan unjuk diri di dunia digital, edukasi, perdagangan/bisnis,

Inovasi di abad digital melahirkan bisnis bidang teknologi informasi yang makin dibutuhkan. Hampir semua sektor pemerintahan dan bisnis mengubah cara berbisnis dengan menyediakan jasa layanan online. Anak-anak muda yang kreatif menumbuhkan perusahaan-perusahaan dari garasi, yang melahirkan perusahaan raksasa di bidang teknologi Informasi seperti Microsoft, Apple, Google, dan Facebook. Industri berbasis Teknologi Informasi tumbuh pesat, dengan yang terkenal misalnya Silicon Valley. Berbeda dengan model bisnis biasanya, model bisnis di bidang teknologi informasi ini mampu mendominasi akibat adopsi pasar yang sangat cepat. Suatu bisnis media sosial misalnya dapat memperoleh pengguna hingga miliaran hanya dalam waktu kurang dari 5 tahun. Perkembangan ini mendisrupsi bursa perusahaan terkemuka di dunia yang saat ini menjadi didominasi oleh perusahaan berbasis teknologi informasi. Jika pada tahun 2013, lima perusahaan dengan pasar terbesar terdiri atas Apple (Teknologi), Exxon (Minyak dan Gas), Berkshire Hathaway (Konglomerasi Investasi), PetroChina (Minyak dan Gas), serta Walmart (Retail), pada tahun 2018, posisi lima besar tersebut semuanya dikuasai oleh perusahaan di bidang teknologi informasi seperti Apple, Alphabet, Microsoft, Amazon, dan Tencent.



Gambar 8.5 Ilustrasi Ekonomi Kreatif di Indonesia

Pertumbuhan yang pesat tersebut diakibatkan dari nilai tambah dan manfaat, atau biasa disebut *value*, dari teknologi informasi terhadap seluruh aspek kehidupan manusia. Kemudahan dan keterhubungan yang dimungkinan dengan teknologi informasi telah menjadi kebutuhan primer bagi sebagian besar manusia. Bidang usaha mengadopsi secara masif teknologi informasi agar memenuhi *competitive necessity* yang mereka perlukan agar usaha mereka tidak tersaingi atau dikalahkan oleh perusahaan lainnya. Di sisi lain, investasi di aspek teknologi informasi pun mereka lakukan untuk mendapatkan teknologi yang belum dimiliki oleh saingan mereka sehingga menjadi suatu *competitive advantage* yang dapat membuat mereka unggul dari perusahaan saingan mereka. Saat kalian membaca berita dan menyimak bahwa ada perusahaan yang menginvestasikan sejumlah besar dana pada perusahaan yang memiliki teknologi, hal itu pada dasarnya dilakukan agar mereka dapat memperoleh value yang diberikan oleh perusahaan tersebut dengan lebih cepat.

Di sisi konsumen, *value* tersebut dirasakan di berbagai aspek kehidupan. Pada dasarnya, *value* tersebut berupa suatu kemudahan dan kenyamanan untuk melakukan suatu aktivitas, serta kemudahan untuk berhubungan atau berkomunikasi dengan pihak yang menawarkan layanan. Pada sisi hubungan sesama manusia, media sosial menjadi tidak terpisahkan dengan kehidupan. Misalnya, mulai dari tingkat keluarga, teman, hingga pekerjaan memanfaatkan grup media sosial untuk berkomunikasi hingga berkolaborasi terlepas dari tempat dan waktu. Transaksi yang makin mudah dan variatif dengan adanya perdagangan daring (*e-commerce*) memudahkan kita memperoleh barang atau

jasa yang kita inginkan. Bagi sisi pedagang pun, hal ini membuat mereka dapat menjangkau konsumen di belahan dunia yang lain. Mengadakan perjalanan pun makin mudah dengan banyaknya penyedia layanan digital untuk membeli tiket perjalanan dan hotel, lengkap dengan ulasan dari pengunjung-pengunjung sebelumnya. Pada akhirnya, muncul beberapa aplikasi yang menyediakan beragam layanan hanya dalam satu aplikasi yang disebut aplikasi super (*super app*).

Melihat peluang tersebut, berbagai perusahaan rintisan (startup) hadir di Indonesia. Data dari Badan Ekonomi Kreatif, yang sekarang telah bergabung dengan Kementerian Pariwisata dan Ekonomi Kreatif, menyebutkan bahwa ada 992 perusahaan rintisan di bidang teknologi di Indonesia pada tahun 2018. Sebanyak 352 berada di bidang perdagangan elektronik, 53 berada di bidang teknologi finansial, 55 di bidang *video game,* dan 532 lainnya tersebar di berbagai kategori. Pada tahun 2016, dari pendapatan domestik bruto sebesar 922,59 triliun rupiah, subsektor aplikasi dan *video game* menyumbang 1.86% dari jumlah tersebut. Di wilayah Asia Tenggara, yang memiliki Unicorn (nilai valuasi di atas 1 miliar dolar Amerika) terbanyak di wilayah Asia, Indonesia melahirkan empat dari delapan perusahaan dengan nilai valuasi terbesar, yaitu Go-Jek, Traveloka, Tokopedia, dan Bukalapak. Pemerintah Republik Indonesia, melalui berbagai badan, membuat banyak program untuk melahirkan lebih banyak lagi perusahaan rintisan dan juga meningkatkan jumlah talenta digital yang siap menjadi penggeraknya.

Nilai valuasi, atau 'harga', dari suatu perusahaan rintisan dilihat dari value yang mereka berikan, berbeda dengan bisnis pada umumnya yang melihat dari seberapa besar keuntungan yang mereka hasilkan pada suatu periode. Beberapa indikator dari besarnya *value* tersebut dapat dilihat jumlah pengguna layanan tersebut dan jumlah investasi yang diberikan oleh para investor pada perusahaan rintisan tersebut. Nilai valuasi tersebut menjadi sebuah ukuran untuk yang dapat digunakan untuk menjual kepemilikan atas perusahaan rintisan tersebut. Kalian mungkin pernah mendengar cerita tentang suatu perusahaan yang mengakuisisi atau membeli perusahaan rintisan. Proses akuisisi tersebut dilakukan dengan melihat nilai valuasi tersebut. Selain keuntungan yang dihasilkan, dari valuasi inilah para pendiri perusahaan rintisan, atau biasa disebut founder, mendapatkan keuntungan.

Para pendiri ini melahirkan perusahaan tersebut berdasarkan hasil pemikiran mereka saat melihat fenomena dan permasalahan yang terjadi di masyarakat. Mereka mengaitkan hal tersebut dengan potensi yang diberikan oleh teknologi informasi untuk menyelesaikan permasalahan tersebut.

Misalnya, mengapa ojek online lahir? Karena penggagasnya berpikir untuk memberi solusi bahwa orang perlu cepat, sedangkan di kota, seringkali banyak kemacetan. Dengan sepeda motor, akan lebih cepat. Supaya lebih mudah, pemesanannya online, tidak perlu pergi ke pangkalan ojek. Tukang ojek pun dapat berada di mana pun untuk menerima order. Itu semua merupakan hasil pemikiran. Oleh karena itu, ‘hasil pemikiran intelektual’ tersebut perlu dilindungi secara legal agar tidak menimbulkan sengketa. Perlindungan ini diberikan dalam bentuk hak kekayaan intelektual. Hak kekayaan intelektual (*intellectual property rights*) adalah hak yang timbul dari hasil olah pikir yang menghasilkan suatu produk atau proses yang berguna untuk manusia. Pada intinya, kekayaan intelektual adalah hak untuk menikmati secara ekonomis hasil dari suatu kreativitas intelektual, termasuk software.

Hasil kreasi dari berpikir itu menimbulkan inovasi, jika diwujudkan, dapat menjadi produk-produk yang bernilai bagi masyarakat dan bernilai ekonomi serta mengangkat martabat bangsa, merupakan suatu pengindustrian inteligensi. Bidang informatika sangat berpotensi untuk berkontribusi dalam pengindustrian inteligensi, dengan menghasilkan produk-produk yang meningkatkan berfungsinya sains dan teknologi di dalam tata kehidupan bermasyarakat, yang akan membawa masyarakat Indonesia lebih mampu mewujudkan kemerdekaan berbangsa dan bernegara (Saswinadi 2017).

Ayo Berdiskusi

Aspek Ekonomi Produk Informatika

Pada bagian ini, kalian akan berdiskusi secara berkelompok pada satu atau lebih topik berikut. Ikutilah instruksi dari guru di kelas.

1. Carilah di internet, berbagai “gerai” digital yang disebutkan di atas. Golongkanlah sesuai area bisnisnya (misal: transportasi, travel, finansial, kesehatan, makanan dan minuman, jasa, dll). Rangkum dalam sebuah tabel berikut. Perhatikanlah, bahwa sesuai dengan namanya, biasanya, perusahaan memulai dengan satu area bisnis tertentu. Dalam perjalanannya, mereka membuka layanan “satu pintu” untuk berbagai transaksi lainnya. Menurut kalian, mengapa mereka melakukan hal itu?

Nama Perusahaan	Daftar Layanan	Nama Deskripsi Layanan Utamanya

2. Beberapa anak muda yang sering bepergian ke luar negeri menerima jasa titipan belanja. Mereka menawarkan jasanya secara online lewat media sosial. Berkat itu, mereka bisa menikmati lebih sering bepergian "gratis", dan yang menitip pun senang karena mendapatkan barang yang sulit diperoleh di Indonesia dengan mudah tanpa harus bepergian ke luar negeri yang ongkosnya mahal, dan untuk barang tertentu bisa bebas pajak. Media sosial yang semula dimaksudkan untuk berkomunikasi dan bergaul di dunia maya menjadi sarana bisnis. Menurut kalian, apakah hal ini sah-sah saja?
3. Generasi muda zaman sekarang makin sedikit membaca koran. Di zaman dulu, setiap pagi, orang menanti kedatangan pengantar yang membawa koran ke rumah, atau membeli koran yang ditawarkan di perempatan jalan. Beberapa koran sudah beralih ke koran digital, dan selain tetap menjual koran dalam media cetak, mereka membuka situs internet menyediakan layanan berita "berbayar" dan juga ada yang secara "gratis". Menurut kalian, bagaimana masa depan koran dan profesi pengantar koran tersebut?
4. Pernahkah kalian membuka aplikasi untuk menonton video atau musik? Aplikasi untuk menonton video tersebut memungkinkan kalian menonton video yang "dititipkan" di situs mereka, dan jika dibuka publik, siapa saja dapat membuka video kalian. Dengan aplikasi pemutar video, kalian dapat menikmati berita, belajar, nonton film hiburan, dan sebagainya. Namun, sering kali di awal, ditayangkan iklan yang harus dibaca dan hanya dapat dilewati (skip) setelah beberapa detik. Berikan pendapat kalian:
    a. Apakah kalian "terganggu" dengan iklan yang muncul tanpa kalian minta?
    b. Mengapa ada iklan tersebut? Andaikata kalian pemilik situs video atau berita, mengapa kalian mengizinkan perusahaan lain menayangkan iklan di lapak kalian?
    c. Tanyakan orang tua atau keluarga kalian yang berasal dari generasi sebelumnya. Mereka tidak mengenal aplikasi video dan jika ingin menonton film, mereka pergi ke gedung bioskop. Adakah iklan yang diputar pada saat menonton film?
    d. Seandainya kalian ingin menonton tanpa iklan sama sekali, mungkinkah mematikannya, bagaimana caranya?
5. Selain koran, buku juga makin banyak dalam bentuk elektronik. Sama dengan perangkat lunak, ini menimbulkan masalah karena dengan mudah dapat dibagikan *file*-nya. Jika dalam perpustakaan dunia nyata

buku dapat dikembalikan, tidak demikian halnya dengan beberapa perpustakaan digital yang dioperasikan. Menurut kalian, apa yang paling perlu diperhatikan, di mana kita hidup dalam sebuah dunia digital, yang sekali barang diberikan tak dapat ditarik kembali? Dalam kaitannya dengan aspek ekonomi, apa pendapat kalian yang perlu dilakukan?

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Hal baru apa yang kalian pelajari dari aktivitas hari ini?
2. Apakah sekarang kalian memiliki keinginan untuk menciptakan inovasi berbasis informatika untuk meningkatkan kegiatan ekonomi?
3. Inovasi informatika apa yang terpikirkan oleh kalian setelah melaksanakan aktivitas ini?

2. Aspek Hukum

Perangkat lunak merupakan hasil buah pikir (karya intelektual) manusia, seperti halnya buku atau karya lainnya. Oleh karena itu, perangkat lunak harus dihargai sebagai salah satu hak kekayaan intelektual. Sayangnya, karena bentuknya yang berupa benda digital, perangkat lunak lebih mudah untuk diperoleh melalui cara yang tidak legal/benar secara hukum, atau dengan istilah umumnya ialah melalui pembajakan. Perlu ditekankan bahwa melakukan pembajakan perangkat lunak itu tindakan melawan hukum.

Untuk dapat memahami hal-hal yang boleh/tidak boleh dilakukan terhadap suatu perangkat lunak, kita perlu melihat jenis lisensi apa yang disediakan oleh pembuat perangkat lunak tersebut. Lisensi perangkat lunak mencakup izin, hak, dan pembatasan yang diberlakukan atas perangkat lunak, baik berupa suatu komponen atau program berdiri sendiri. Penggunaan suatu perangkat lunak tanpa lisensi dapat dianggap pelanggaran atas hak eksklusif pemilik menurut hukum hak cipta atau, kadang, paten dan dapat membuat pemilik menuntut pelanggarnya. Dalam suatu lisensi, penerima lisensi diizinkan untuk menggunakan perangkat lunak berlisensi sesuai dengan persyaratan khusus dalam lisensi. Pelanggaran persyaratan lisensi, bergantung pada lisensinya, dapat menyebabkan pengakhiran lisensi, dan hak pemilik untuk menuntut pelanggarnya. Lisensi dapat dianggap sebagai sebuah perjanjian hukum yang resmi antara pembuat perangkat lunak yang menyediakan lisensi tersebut (*licensor*) dan pihak pengguna perangkat lunak yang menerima lisensinya (*licensee*).

Ada beberapa jenis lisensi perangkat lunak. Berikut ini rangkuman beberapa jenis utama dari lisensi perangkat lunak.

a. Lisensi Komersial

Lisensi komersial adalah lisensi yang bersifat paling restriktif (mengikat). Sesuai namanya, lisensi ini biasanya diterapkan untuk perangkat lunak yang berbayar. Pengguna hanya diperbolehkan menggunakan perangkat lunak setelah membayar suatu harga tertentu kepada pembuat perangkat lunak. Selain itu, dalam penggunaannya pun, biasanya, terdapat berbagai peraturan yang mengatur hal yang boleh/tidak boleh dilakukan oleh pengguna selama masa pemakaian, misalnya: hanya boleh meng-instalasi dalam jumlah tertentu atau pada perangkat tertentu, tidak boleh memperbanyak/menggandakan perangkat lunak di luar yang telah disepakati, tidak boleh mengubah kode program atau perilaku perangkat lunak di luar metode yang diperbolehkan, dan lain-lain. Perangkat lunak dengan lisensi komersial ini tentunya rentan terhadap tindakan pembajakan, terutama perangkat lunak yang diperlukan oleh banyak orang, seperti: sistem operasi, pengolah kata, pengolah gambar dan lain-lain.

b. Lisensi Gratis (*Freeware*)

Sebagian perangkat lunak mungkin disediakan secara gratis, dan dapat diunduh serta di-*install* dari sebuah sumber tertentu yang disediakan oleh penggunanya. Meskipun perangkat lunak tersebut disediakan secara gratis, bukan berarti tidak ada batasan yang diatur dalam lisensinya. Misalnya, meskipun sebuah perangkat lunak berlisensi gratis, tetapi mungkin tidak memperbolehkan pengguna untuk mendistribusikan ulang perangkat lunak tersebut, atau mengubah kode programnya. Sebuah variasi lisensi yang mirip (tetapi tidak sama persis) dengan lisensi gratis ialah lisensi *shareware*, yang memperbolehkan pengguna untuk secara gratis mencoba perangkat lunak tersebut selama masa percobaan tertentu yang ditetapkan (misal selama 1 bulan), dan setelah itu, pengguna diminta untuk membayar untuk menggunakan perangkat tersebut lebih lanjut.

c. Lisensi *Open Source*

Lisensi *open source* adalah lisensi yang memperbolehkan pengguna untuk tidak hanya menggunakan perangkat lunak tersebut, tetapi juga untuk melihat, mengubah dan mendistribusikan kode sumber program yang digunakan untuk membuat perangkat lunak tersebut. Dengan menggunakan lisensi ini, penulis perangkat lunak dapat membagikan kode program yang dibuatnya, agar dapat ditingkatkan oleh orang lain sehingga dapat

menimbulkan kolaborasi/kerja sama untuk membangun sebuah perangkat lunak berkualitas yang dapat digunakan oleh khalayak umum secara gratis (tanpa berbayar). Saat ini, *open source* sudah menjadi sebuah gerakan yang didukung oleh banyak orang, organisasi, perusahaan serta pemerintahan di seluruh dunia, dan telah mampu menghasilkan berbagai perangkat lunak berkualitas, mulai dari sistem operasi (misal Linux), perangkat pengolah dokumen (*office*), penjelajah internet (*browser)*, klien email, pengolah gambar, pemutar dokumen multimedia (suara dan video), dan lain sebagainya. Salah satu pelopor gerakan *open source* ialah proyek GNU (GNUis Not Unix) yang mulai memopulerkan sistem operasi yang gratis. Pada pertengahan 1980-an, proyek GNU mengeluarkan lisensi-lisensi perangkat lunak bebas yang terpisah untuk setiap paket perangkat lunaknya. Kesemuanya digantikan pada 1989 dengan versi satu dari Lisensi Publik Umum GNU (GNU General Public License disingkat GPL). Versi 2 dari GPL yang dirilis pada 1991, dan diikuti dengan dengan GPL versi 3 pada tahun 2007.

d. Lisensi Domain Publik

Lisensi adalah jenis lisensi di mana semua hak cipta terhadap perangkat lunak telah dilepaskan (oleh si pembuat perangkat lunak) sehingga kepemilikan perangkat lunak tersebut diberikan kepada masyarakat umum. Setiap orang diperbolehkan untuk menggunakan, menggandakan, dan mendistribusikan ulang, ataupun mengubah kode programnya tanpa ada batasan.



Gambar 8.6 Komik tentang Seseorang yang Sedang Membeli Komputer

Lisensi perangkat lunak bebas memberikan banyak kemajuan di bidang pengembangan perangkat lunak dan juga perangkat keras karena adanya keterbukaan informasi tentang *software*. Perdebatan mengenai hak kekayaaan intelektual perangkat lunak dikaitkan dengan kebebasan memakai dan memodifikasi ini masih akan menjadi perdebatan yang tidak selesai, tetapi akan membawa banyak kemajuan dan inovasi.

Yayasan Perangkat Lunak Bebas (FSF, fsf.org), sebuah organisasi yang mengeluarkan Definisi Perangkat Lunak Bebas, merilis daftar lisensi perangkat lunak bebas. Daftar tersebut membedakan antara lisensi perangkat lunak bebas yang kompatibel dan yang tidak kompatibel dengan lisensi pilihan FSF, yaitu Lisensi Publik Umum GNU, yang merupakan sebuah lisensi copyleft. Dalam daftar tersebut, juga tercantum lisensi-lisensi yang dianggap FSF tidak bebas untuk beberapa alasan tertentu.

Di Indonesia, kesadaran masyarakat tentang lisensi perangkat lunak masih perlu ditingkatkan. Masih sering kita temui adanya pelanggaran terhadap lisensi perangkat lunak, baik melalui pembajakan, penggandaan perangkat lunak yang tidak sesuai dengan lisensi, serta penjualan perangkat lunak tanpa lisensi resmi. Padahal, sudah ada undang-undang yang mengatur mengenai hal ini. Di Indonesia, lisensi perangkat lunak tercakup dalam konsep Hak Kekayaan Intelektual (HaKI), yang diatur dalam Undang-Undang No. 7 Tahun 1994 tentang Pengesahan WTO (Agreement Establishing The World Trade Organization), dan lebih spesifik lagi pada Pasal 40 Undang Undang No. 28 Tahun 2014 tentang Hak Cipta (UU Hak Cipta). Istilah HaKI atau Hak atas Kekayaan Intelektual merupakan terjemahan dari *Intellectual Property Right* (IPR), yang memiliki pengertian pemahaman mengenai hak atas kekayaan yang timbul dari kemampuan intelektual manusia, yang mempunyai hubungan dengan hak seseorang secara pribadi, yaitu hak asasi manusia (human right). HaKI adalah hak eksklusif yang diberikan suatu hukum atau peraturan kepada seseorang atau sekelompok orang atas karya ciptanya.

Untuk memperbaiki kondisi penghargaan terhadap hak cipta terkait penggunaan perangkat lunak di Indonesia, perlu kita pahami dan tumbuhkan kesadaran mengenai pentingnya mematuhi aspek legal lisensi perangkat lunak dalam kehidupan sehari-hari. Banyak orang yang dalam kehidupan sehari-harinya tidak/jarang melanggar hukum, tetapi dalam penggunaan perangkat lunak, mereka tidak begitu memperdulikan apakah perangkat lunak yang digunakan memiliki lisensi yang benar/tidak. Padahal, dari kacamata hukum, tindakan menggunakan perangkat lunak dengan lisensi yang tidak legal sama saja dengan tindakan melanggar hukum lain, misalnya mencuri. Oleh karena itu, perlu ditekankan pemahaman bahwa pelanggaran lisensi perangkat lunak juga adalah kegiatan melawan hukum yang harus dihindari.

Alasan lain yang juga menjadi penyebab mengapa masih banyak yang menggunakan perangkat lunak tanpa lisensi yang benar ialah permasalahan biaya: banyak perangkat lunak populer yang diperlukan memiliki lisensi komersial dan dijual dengan harga yang tidak murah dan masih memberatkan bagi banyak orang di Indonesia. Sebagai salah satu solusi permasalahan ini

ialah dengan meningkatkan kesadaran dan pengetahuan rakyat Indonesia akan adanya berbagai alternatif perangkat lunak dengan lisensi gratis ataupun *open source* yang memiliki kemampuan dan fungsionalitas yang tidak kalah dari perangkat lunak komersial, dan dalam banyak hal sudah mencukupi untuk kebutuhan sebagian besar pengguna.

Ayo Berdiskusi

Aspek Hukum Produk Informatika

1. Di dunia nyata, banyak barang bermerk yang mahal harganya ditiru, seperti tas, koper, baju, dan sebagainya. Pembuat barang palsu mencantumkan merk yang sama pada barang yang diproduksinya. Hal ini juga terjadi pada perangkat lunak atau produk dalam bentuk digital lainnya, yang secara populer biasa disebut perangkat lunak/video/buku “bajakan”. Menurut kalian, apa bedanya membajak barang nyata dan barang digital? Apakah tindakan membajak itu boleh dilakukan?
2. Banyak perangkat lunak disediakan untuk dapat dipakai secara legal dan “gratis”. Tentu, kalian sudah menikmati beberapa perangkat lunak semacam itu. Dari aspek ekonomi perangkat lunak, kalian tahu bahwa mengembangkan perangkat lunak membutuhkan biaya. Menurut kalian, bagaimana perusahaan pengembang perangkat lunak bisa hidup?
3. Carilah di internet, perangkat lunak berlisensi dan “padanan”-nya (artinya yang fungsi utamanya sama, walau mungkin tidak sama persis).

No	Jenis Perangkat Lunak	Merk Berlisensi	Gratis
1	Sistem Operasi Smartphone		
2	Sistem Operasi PC		
3	Paket Office		
4	Pengolah kata		
5	Pengolah lembar kerja		
6	Pengolah Presentasi		
7	Editor gambar		
8	Kompiler Bahasa Pemrograman C		
9	Pemutar media (suara, video)		

4. Carilah di internet halaman web atau buku yang dapat diunduh dengan berlisensi gratis, misalnya buku di https://csunplugged.org. Lisensinya disebut CCL (Common Creative Licence). Apa saja yang boleh dilakukan dengan buku yang kalian *download* dengan lisensi tersebut? Ada berapa jenis lisensi CCL?

5. Carilah referensi di internet (Wikipedia, Blog, Berita, pencarian di Google) untuk dapat menjawab pertanyaan berikut: “Apa yang menjadi motivasi bagi orang-orang ataupun perusahaan-perusahaan tertentu untuk menulis sebuah program untuk perangkat lunak, lalu kemudian menerbitkan perangkat lunak tersebut dengan lisensi *open source* sehingga orang lain dapat menggunakan perangkat lunak tersebut tanpa membayar?”
6. Ada beberapa jenis lisensi open source yang berbeda dengan konsekuensi/ akibat yang berbeda-beda juga bagi pengguna perangkat lunak dengan lisensi tersebut (baik pengguna yang sekadar menggunakan ataupun yang akan memodifikasi kode program perangkat lunak tersebut). Contoh dua buah lisensi *open source* yang cukup berbeda ialah GPL (v1 – v3) dan BSD. Carilah referensi di internet untuk dapat memahami, apa perbedaan utama antara kedua jenis lisensi open source tersebut!

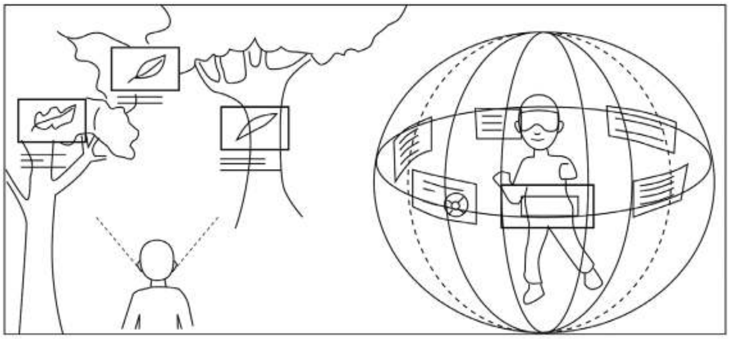
Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Hal baru apa yang kalian pelajari dari aktivitas hari ini?
2. Dari pelajaran hari ini, bagaimana kalian menyikapi pembajakan perangkat lunak?
3. Apakah kalian menjadi tertarik untuk menghasilkan karya intelektual yang dapat digunakan secara bebas oleh orang lain, demi manfaat bagi orang banyak? Mengapa?

C. Informatika untuk Masa Depan

Kreativitas dan inovasi berawal dari berpikir dan berimajinasi. Banyak penulis dan sastrawan yang berani mengungkapkan bayangan mereka akan masa depan. Banyak yang membayangkan seperti apa dunia di abad ke-21, bahkan lebih jauh lagi ketika manusia sudah melintasi galaksi. Imajinasi itu tidak lagi menjadi sekadar hiburan, tetapi akhirnya menjadi sumber inspirasi bagi para inovator dan penemu yang perlahan-lahan mengubah dunia dengan segala teknologi mutakhir yang mereka ciptakan. Nah, kini, tibalah saatnya kalian juga mulai berimajinasi dan membayangkan seperti apa negeri kita ini, NKRI – Negara Kesatuan Republik Indonesia, di masa depan.



Gambar 8.7 Inovasi-Inovasi Unggul di Bidang Informatika Banyak Bermula dari Coretan-Coretan Kreatif Di Buku Catatan atau Jurnal Seseorang.

Republik Indonesia, sebagai sebuah negara berdaulat, harus terus tumbuh dan memantapkan posisinya di dunia internasional untuk mewujudkan dan mempertahankan suatu negara yang adil dan makmur. Perserikatan Bangsa-Bangsa (PBB) telah mendeklarasikan suatu konsep bernama tujuan pembangunan berkelanjutan (*sustainable development goals*) atau SDG yang berisi tujuan yang harus dicapai pada tahun 2030 agar suatu bangsa dapat menjadi bangsa yang maju secara berkelanjutan.

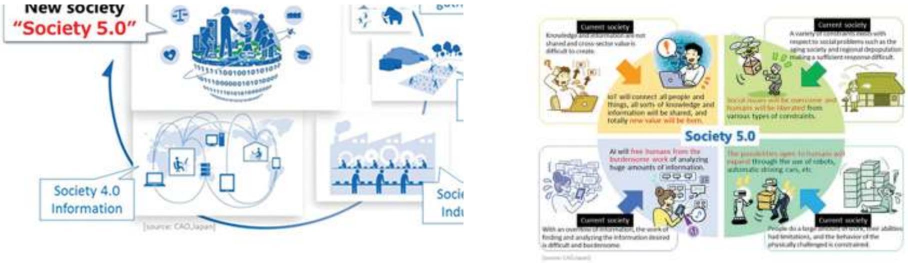
SDG merupakan kelanjutan dari tujuan pembangunan milenium (*millennium development goals*) atau MDG. Akan tetapi, dalam pelaksanaannya, MDG memiliki banyak kendala. Menurut Jeffrey Sachs, salah satu penasihat MDG, salah satu kendala dalam terwujudnya MDG ialah data yang sulit dikumpulkan dan diolah dengan cepat menjadi informasi yang dapat digunakan oleh para pengambil kebijakan. MDG telah berakhir pada tahun 2015, dan kini telah banyak kemajuan pada bidang komputer yang dapat kita gunakan untuk memastikan SDG dapat dicapai dengan baik dan sesuai dengan target.



Gambar 8.8 Tujuan Pembangunan Berkelanjutan

Selain gagasan SDG, masyarakat saat ini telah mengalami kemajuan dengan adanya gagasan Society 5.0, yang didefinisikan sebagai sebuah masyarakat yang berpusat pada manusia, yang menyeimbangkan kemajuan ekonomi dan resolusi masalah sosial. Masyarakat 5.0 dengan kuat mengintegrasikan ruang siber (alam digital, interkoneksi melalui internet) dan ruang fisik (alam raya dan tempat-tempat di bumi yang dibagi menjadi 5 benua dengan lokasi

geografis masing-masing). Masyarakat 5.0 diusulkan pada “5th Science and Technology Basic Plan” sebagai masyarakat masa depan yang menjadi aspirasi Jepang. Masyarakat 5.0 ini merupakan kelanjutan dari masyarakat pemburu (Society 1.0), masyarakat pertanian (Society 2.0), masyarakat industri (Society 3.0), dan masyarakat informasi (Society 4.0).



(a) (b)

Gambar 8.9 Ilustrasi Gambaran Dunia di Masa Depan, Masyarakat 5.0 (Society 5.0) (a) abstraksi, (b) contoh aktivitas

Ayo Berdiskusi

Gagasan Kalian untuk Indonesia di Masa Depan

Saatnya melahirkan gagasan kreatif kalian! Kalian dapat memilih salah satu tujuan pembangunan berkelanjutan, kemudian gambarkanlah gagasan kalian tentang bagaimana komputer dapat digunakan di bidang tersebut di Indonesia pada tahun 2045, untuk mewujudkan Masyarakat 5.0. Kalian dapat membuat rancangan dari gagasan kalian dalam bentuk apa pun, bahkan dapat berupa coretan sketsa yang ditambah dengan penjelasan atau deskripsi. Kemudian, kalian akan mempresentasikan gagasan tersebut di depan kelas. Guru kalian mungkin akan mengadakan pameran kecil untuk menunjukkan gagasan kalian ini ke semua sekolah. Jangan lupa untuk memikirkan seperti apa dampak dari gagasan kalian, baik positif maupun negatif, kepada masyarakat.

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Apakah kalian telah memahami bagaimana komputer dapat kita gunakan untuk mewujudkan Indonesia yang lebih maju secara berkelanjutan?
2. Apakah kalian mengalami kesulitan dalam membangun gagasan kalian? Jika ya, apa yang membuat kalian merasa kesulitan?

3. Gagasan apa yang menurut kalian paling menarik untuk diwujudkan?
4. Apakah diskusi ini membuat kalian tertarik untuk mewujudkan gagasan kalian?

D. Karier dan Studi Lanjut di Bidang Informatika

Di masa kini, dan masa yang akan datang, penerapan informatika di berbagai bidang adalah sebuah keniscayaan. Saat ini, sudah tidak aneh apabila ada seorang yang berkarier di bidang pemasaran menguasai analisis data dalam bentuk intelegensi bisnis, atau seorang ahli biologi yang mengolah data genetika dengan membuat program komputer. Suatu konsep Masyarakat 5.0 pun dicetuskan yang intinya ialah di masa mendatang, profesi apa pun akan berhubungan erat dengan dunia siber dalam menyelesaikan setiap permasalahan dalam kehidupan. Oleh karena itu, penguasaan aspek berpikir komputasional akan menjadi sangat penting dikuasai siswa sejak masih sekolah, terlepas dari profesi yang akan ditekuninya, walaupun pada mata pelajaran informatika ini, studi kasus yang disampaikan terbatas di bidang informatika. Di masyarakat, istilah informatika kurang populer, masih sering disamakan dengan TIK dan Teknologi Informasi. Banyak orang mengatakan: ”Saya bekerja di bidang TI (Teknologi Informasi)” atau guru mengatakan: “Saya mengajar TIK”. Kita perlu meluruskan perbedaan TIK, TI, dan Informatika. TIK lebih berorientasi ke pemakaian aplikasi dan komputer saja. TI memang lebih luas cakupannya. Informatika ialah disiplin ilmu, padanan dari ilmu komputasi atau “computing”. Oleh karena itu, mata pelajaran ini disebut Informatika. TIK akan dibutuhkan untuk memudahkan kehidupan masyarakat digital berkomunikasi lewat teknologi digital. Teknologi informasi menjadi istilah umum dalam berbagai profesi terkait informatika.

Bagi murid kelas X, yang baru akan memasuki dunia kerja sekitar tujuh tahun yang akan datang, perkembangan informatika dan integrasinya dalam berbagai sendi kehidupan akan makin masif. Dalam satu dekade terakhir, ada banyak profesi dan keahlian baru yang lahir dan dibutuhkan oleh industri. Hal demikian juga akan terjadi di masa datang, dengan makin meleburnya teknologi seperti kecerdasan buatan (*artificial intelligence*), pembelajaran mesin (*machine learning*), realitas virtual (*virtual reality*), *internet of things*, dan teknologi lainnya dalam kehidupan. Hal ini didorong dengan perkembangan teknologi yang telah menjadi suatu kebutuhan untuk bersaing (*competitive necessity*) bagi industri untuk dapat unggul pada masa ini. Oleh karena itu, melalui unit ini, murid perlu dikenalkan dengan kompetensi Informatika

yang dapat diperoleh baik melalui studi lanjut atau jalur lain yang diakui lewat proses sertifikasi. Kompetensi tersebut akan bermanfaat untuk meniti karier di masa depan.

1. Karier di Bidang Informatika

Mereka yang bekerja di bidang informatika jarang sekali mengatakan bahwa mereka bekerja di bidang informatika. Biasanya, mereka lebih cenderung menyebut mereka memiliki karier di bidang teknologi informasi atau yang dalam bahasa Inggris disebut *information technology* (IT). Belakangan, ada banyak profesional yang menggunakan berbagai macam posisi pekerjaan yang lebih spesifik seperti *system analyst, back-end engineer, front-end engineer, dev-ops engineer, user experience designer, data scientist, product manager, hingga chief technology officer.* "Posisi" tersebut terus berkembang dan berevolusi mengikuti perkembangan teknologi dan kebutuhan industri yang sedemikian cepatnya.



Gambar 8.10 Ilustrasi Beberapa Profesi di Bidang Informatika

Pada saat ini, kebanyakan orang awam memahami Informatika sebagai disiplin ilmu yang mempelajari fenomena di sekitar komputer. Fenomena ini meliputi desain komputer dan proses komputasi, representasi objek informasi dan transformasinya, perangkat keras, perangkat lunak, efisiensi, dan kecerdasan mesin. Di Eropa, disiplin ilmu komputer disebut "informatika". Di Amerika Serikat, disebut *computing*. Pada unit ini, kita akan melihat lebih

dalam mengenai makna dari suatu profesi di bidang Informatika, yang lebih luas daripada sekadar “mencari nafkah”, serta pengelompokannya di dalam dunia keprofesian.

Menurut Prof. Peter J Denning, seorang peneliti terkemuka yang banyak melahirkan tulisan ilmiah terkait keprofesian di bidang Informatika, suatu profesi harus memiliki empat ciri yang khas. Keempat ciri tersebut seperti berikut.

1. Mendalami suatu bidang yang menjadi perhatian manusia dalam waktu yang lama. Profesi informatika berperan untuk memenuhi kebutuhan manusia atas kemampuan komputasi dan komunikasi yang efektif. Hal ini juga didorong oleh perkembangan teknologi yang makin cepat dan makin melebur dengan seluruh sendi-sendi kehidupan manusia. Pada bidang ini, seorang profesional di bidang informatika akan berusaha menyelesaikan berbagai permasalahan dengan memanfaatkan kemampuan komputer.
2. Mengodifikasikan kumpulan prinsip atau pengetahuan konseptual. Kumpulan prinsip-prinsip ini diwujudkan dalam berbagai bentuk. Kurikulum yang dibuat oleh ACM dan IEEE, program studi di perguruan tinggi, dan lembaga-lembaga pelatihan merupakan salah satu wujud kodifikasi prinsip-prinsip di bidang informatika yang dapat diakses oleh seorang profesional di bidang informatika.
3. Mengodifikasikan kumpulan praktik, yang juga meliputi kompetensi. Hal ini salah satunya diwujudkan dan diakui melalui sertifikasi kompetensi profesional yang telah dijelaskan pada unit sebelumnya.
4. Menetapkankan standar kompetensi, etika, dan praktik. Hal ini menjadi standar bagi seorang profesional di bidang informatika untuk berperilaku dan bersikap dalam menjalankan tugas-tugas profesinya. Hal ini bermanfaat agar profesional di bidang informatika memiliki integritas dan dapat dipercaya.

Berdasarkan uraian di atas, apabila kalian ingin menjadi seorang profesional di bidang informatika, keempat hal tersebut perlu kalian pahami. Pekerjaan bukan sekadar sebagai sarana untuk mendapatkan penghasilan, tetapi untuk memberikan manfaat bagi masyarakat. Selain menguasai kumpulan prinsip dan praktik, kalian pun dapat berpartisipasi aktif untuk merumuskan suatu prinsip atau praktik baru dan ikut menyebarkannya kepada orang lain di profesi yang sama. Hal ini dapat kalian lakukan dengan berbagai cara, seperti menulis, memberikan ceramah, atau membuat pelatihan. Terakhir, kalian harus mengetahui etika serta standar kompetensi dan praktik di bidang itu untuk memastikan kemampuan dan sikap yang kalian tunjukkan menunjukkan integritas dari profesi kalian.

Berdasarkan peran informatika, Prof. Denning membagi profesi di bidang informatika dalam tiga kategori, yaitu profesi yang sangat spesifik di bidang informatika, profesi yang sarat dengan informatika, serta profesi yang memberikan dukungan teknologi informasi. Contoh-contoh pekerjaan tersebut dapat dilihat pada tabel di bawah ini.

Tabel 8.1 Profesi di Bidang Informatika

Profesi yang Sangat Spesifik di Bidang Informatika	Profesi yang Sarat dengan Informatika	Profesi yang Memberikan Dukungan Teknologi Informasi
Kecerdasan Buatan Ilmu Komputer Rekayasa Komputer Ilmu Komputasional Rekayasa Basis Data Grafika Komputer	Dirgantara Bioinformatika Sains Kognitif Sains Perpustakaan Digital E-Commerce Layanan Finansial	Teknisi Komputer Teknisi Helpdesk Teknisi Jaringan Pelatih IT Professional Spesialis Keamanan Siber Administrator Sistem
Interaksi Manusia-Komputer Rekayasa Jaringan Sistem Operasi Rekayasa Kinerja Robotika Komputasi Saintifik Arsitektur Perangkat Lunak Rekayasa Perangkat Lunak Keamanan Sistem	Rekayasa Genetika Ilmu Informasi Kebijakan Publik Privasi	Desainer Layanan Web Desainer Identitas Web Administrator Basis Data

Setiap bidang profesi akan dikaitkan dengan profesional (pekerjanya). Secara umum, pekerjaan di bidang informatika akan dikelompokkan ke bidang ilmu yang sesuai dengan bidang Program Studi yang ditempuh sebelum bekerja. Karena bidang informatika mencakup teori sampai dengan praktek, bidang ilmu dasar sampai dengan teknologi dan aplikasinya, lulusan bidang informatika dapat memilih jalur sains atau teknologi, menjadi ilmuwan (*scientist*), atau menjadi insinyur (*engineer*). Ilmuwan "mengamati" dan mengeksplorasi dunia untuk dapat menjelaskan secara ilmiah fenomena alam yang terjadi, sementara insinyur lebih fokus pada penciptaan produk-produk yang diciptakan untuk menunjang kehidupan dengan memanfaatkan hasil ilmuwan, sekaligus menemukan dan menciptakan ilmu rekayasa yang baru.

Saat ilmuwan perlu membuktikan suatu konsep ilmiah, ia perlu melakukan observasi, analisis serta merancang dan melaksanakan eksperimen. Insinyur menciptakan produk, dengan melakukan tahapan berikut: analisis, perancangan, implementasi, pengujian, pemasangan dan pengoperasian. Pada saat pengoperasian, dibutuhkan pemeliharaan produk. Insinyur dibantu oleh teknisi, dan pemakaian produk membutuhkan operator.

Secara lebih spesifik, profesi di bidang informatika dapat dibagi-bagi menjadi beberapa posisi pekerjaan (*job title*) yang memiliki deskripsi serta tanggung jawab yang berbeda. Posisi-posisi tersebut senantiasa berkembang dan berevolusi. Akibatnya, bisa jadi ada posisi baru yang muncul, berubah, bergabung, atau berkurang. Ketika seseorang melamar pekerjaan, atau berusaha merekrut orang lain untuk bekerja di usaha miliknya, deskripsi dan tanggung jawab ini perlu dicermati.

Secara lebih rinci, beberapa pekerjaan di bidang informatika diberikan pada Tabel 8.2. Selain posisi pekerjaan yang disebutkan ini, masih banyak posisi pekerjaan lain, yang dapat kalian temui di situs-situs penawaran pekerjaan atau media jaringan sosial tempat para profesional berkumpul. Posisi pekerjaannya dituliskan dalam istilah bahasa Inggris.

Tabel 8.2 Jenis-Jenis Profesi di Bidang Informatika

Posisi Pekerjaan	Deskripsi dan Tanggung Jawab
Chief Information Officer (CIO)/Chief Technology Officer (CTO)	Bertanggung jawab pada seluruh aspek terkait teknologi informasi di suatu perusahaan.
Computer Programmer	Menulis program komputer yang telah didesain.
Computer Scientist	Menciptakan suatu prinsip, metode, atau praktik baru di bidang ilmu komputer.
Hardware Engineer	Menciptakan, mengimplementasikan, dan menguji komponen fisik dari komputer.
Software Engineer	Mengembangkan, merawat, menguji, mengevaluasi, dan meningkatkan kualitas suatu perangkat lunak berdasarkan kebutuhan yang telah diberikan oleh analis sistem atau arsitek.
Software Developer	Menganalisis kebutuhan pengguna dan software requirements untuk menentukan kelayakan desain dalam batasan waktu, risiko, kualitas dan biaya. Mengembangkan desain sistem, software, prosedur dan dokumentasi pengujian.
Web Developer	Bertanggung jawab untuk membuat web.
Database Administrator	Mengembangkan dan menerapkan pemantauan, kinerja, kapasitas, dan strategi perluasan database. Menginstal, mengonfigurasi, meng-upgrade, memantau, dan melakukan pemeliharaan pada database.
IT Architect	Mendesain arsitektur teknologi informasi untuk mendukung proses suatu perusahaan.
Network Administrator	Memelihara, mengoperasikan, mengelola, dan mendukung infrastruktur jaringan.

Posisi Pekerjaan	Deskripsi dan Tanggung Jawab
Systems Analyst	Menganalisis dan menerjemahkan tujuan bisnis menjadi produk-produk, dalam konteks Informatika, produk ini disebut computational artefact, yaitu berupa proses bisnis dan spesifikasi kebutuhan sistem informasinya.
Security Analyst	Menganalisis dan menerjemahkan aspek keamanan dari sistem yang akan dikembangkan untuk mendukung tujuan bisnis.
Information Researcher	Menciptakan suatu prinsip, metode, atau praktik baru untuk mengolah data dan informasi.
Video Game Developer	Mengembangkan perangkat lunak permainan dan bekerja sama dengan tim profesional kreatif untuk meningkatkan kualitas permainan.
Health Information Technician	Mengolah data terkait kesehatan menggunakan prinsip dan praktik informatika.
Data Scientist	Merancang, mengembangkan, dan menerapkan model statistik prediktif (biasanya pada big data sets).
Data Engineer	Mengolah data yang dimiliki perusahaan untuk mendapatkan wawasan baru yang berguna bagi perusahaan.
Web Designer	Merancang antarmuka pengguna web untuk memberikan grafik, konten, markup dan skrip.
Mobile Apps developer	Bertanggung jawab mengembangkan aplikasi dalam bentuk mobile.

Adopsi sebuah teknologi baru di dunia industri akan membutuhkan posisi-posisi baru. Pada saat perusahaan mulai mengadopsi sistem informasi untuk membantu pekerjaan mereka, mereka membuka posisi yang terkait dengan rekayasa perangkat lunak, misalnya posisi analis sistem, pemrogram, dan perekayasa basis data. Kemudian, isu keamanan siber meningkat sehingga banyak posisi di bidang analis keamanan siber (*cyber security*) menjadi banyak ditawarkan. Saat muncul tuntutan untuk menghasilkan sebuah perangkat lunak yang mudah digunakan dan memberikan kesan elegan bagi pengguna, posisi yang terkait dengan desain pengalaman pengguna (*user experience*) dan kebergunaan (*usability*) sistem pun ditawarkan. Saat ini, ketika sebuah perangkat lunak diharapkan dapat bekerja dengan cepat walaupun digunakan oleh jumlah pengguna yang sangat besar, banyak usaha pun mencari orang-orang untuk mengisi posisi pengembangan dan operasional IT (*DevOps*) dan arsitek perangkat lunak. Akhirnya, pada saat ini, mulai banyak perusahaan yang merekrut orang-orang dengan kemahiran di analisis data, kecerdasan

buatan, atau pembelajaran mesin untuk melakukan otomatisasi pada proses bisnis di perusahaan tersebut. Contoh yang paling lazim ialah untuk mendeteksi penipuan (*fraud*) pada transaksi secara elektronik.

Ilustrasi di atas terjadi karena banyak badan usaha melakukan transformasi digital dengan menerapkan teknologi informasi dalam berbagai proses mereka lakukan. Sebagian perusahaan tidak mengembangkan dan memelihara sistem, tetapi menyewa jasa SAS (*Software as a service*) yang beroperasi di Cloud. Saat ini, pengembangan maupun penerapan teknologi informasi tersebut bukan lagi suatu hal yang membuat suatu usaha menjadi unggul dalam persaingan (*competitive advantages*), tetapi merupakan suatu syarat agar usaha tersebut dapat bersaing di dunia bisnis (*competitive necessity*). Perubahan ini terjadi dengan sangat cepat, terutama dalam satu dekade terakhir di Indonesia. Oleh karena itu, dalam berkarier di bidang informatika, sangatlah penting untuk selalu meningkatkan kompetensi baik dalam bentuk prinsip maupun praktik untuk menjaga daya saing di kebutuhan dunia yang berubah dengan cepat.

Ketika perusahaan, yang menyadari pentingnya informatika, membuka posisi baru di bidang ini, kebutuhan tenaga kerja bidang informatika pun meningkat. Akan tetapi, menurut Kementerian Komunikasi dan Informatika, jumlah talenta di bidang informatika masih sangat kurang. Bahkan, di tahun 2017, Kemenkominfo menyatakan bahwa Indonesia Darurat Tenaga Programmer. Tidak hanya di Indonesia, ketimpangan tersebut juga terjadi di negara besar seperti Amerika Serikat. Menurut Code.org pada tahun 2020, dari 398,857 lowongan, hanya ada 71,226 lulusan perguruan tinggi yang dapat mengisi posisi tersebut.

Sebagai seorang pelajar di tingkat SMA, persiapan untuk meniti karier di bidang informatika telah dapat dilakukan sejak saat ini. Kalian dapat memanfaatkan media sosial yang sering digunakan untuk bergabung dengan grup-grup komunitas di bidang informatika atau mengikuti kegiatan berbagai atau meet up yang dilaksanakan oleh komunitas atau perusahaan di bidang teknologi secara rutin. Bertemu langsung dengan orang-orang yang sudah berkarier di bidang informatika dapat membuat kalian memahami lebih jauh tentang tanggung jawab profesional yang mereka miliki. Selain itu, kalian pun juga dapat melihat tentang tren teknologi yang sedang berkembang, terutama di Indonesia. Membaca laporan-laporan terkini dari World Economic Forum atau lembaga lain juga akan membantu untuk melihat bidang kerja di masa depan yang kalian minati.

Dengan banyaknya posisi pekerjaan baru yang bermunculan, ingatlah bahwa bisa jadi bidang pekerjaan kalian baru akan muncul setelah kalian lulus dari perguruan tinggi kelak. Di sinilah rasa keingintahuan, inisiatif,

dan kegigihan kalian untuk belajar dan beradaptasi akan membantu kalian menjadi seorang pribadi yang tidak hanya unggul, tetapi juga dapat membawa Indonesia menjadi negara maju dengan kekuatan ekonomi dan inovasi yang besar. Kalian dapat menjadi bagian penting untuk menggapai cita-cita mulia Indonesia Emas pada tahun 2045.

Ayo Lakukan

Mencari Pekerjaan Impian Kalian

Carilah halaman lowongan pekerjaan di salah satu aplikasi yang kalian gunakan sehari-hari. Kalian dapat menggunakan mesin pencari dengan kata kunci pencarian: “Nama Produk Career” atau “Nama Produk + Karier” untuk bahasa Indonesia. Amati posisi-posisi pekerjaan yang ditawarkan beserta dengan deskripsi dan tanggung jawabnya. Dari semua posisi yang kalian baca, yang manakah yang menurut kalian menarik? Apa alasannya? Isilah hasil pekerjaan kalian pada riwayat hidup yang formatnya akan diberikan oleh guru. Kalian tetap bisa membuat riwayat hidup tersebut menjadi lebih menarik dengan kreasi kalian.

2. Studi Lanjut di Bidang Informatika

Disiplin ilmu Informatika adalah disiplin ilmu sains formal yang lahir dari bidang matematika (sebagai konsep formal) dan teknik elektro (sebagai elemen perangkat keras). Saat ini, Informatika telah menjadi suatu bidang ilmu tersendiri dan memiliki berbagai cabang ilmu yang akan dikenalkan pada unit ini. Saat ini, Informatika telah berkolaborasi dengan bidang ilmu lain sehingga melahirkan bidang baru. Agri-Informatika, atau Informatika Pertanian, lahir dengan menerapkan informatika untuk mendukung bidang pertanian. Di bidang kesehatan, lahir Informatika Kesehatan (Health-Informatics), yang memanfaatkan informatika untuk meningkatkan kualitas layanan kesehatan. Di bidang biologi, data genetika diolah dengan menggunakan informatika, yang melahirkan ilmu Bioinformatika. Selain itu, juga terdapat bidang lain seperti Urban Informatics, Travel Informatics, dan Geoinformatika.

Dalam bidang informatika yang telah dipelajari, terdapat lima pilar yang terdiri atas sistem komputer, jaringan komputer dan internet, analisis data, algoritma dan pemrograman, serta TIK dan dampaknya terhadap kehidupan manusia. Dua asosiasi profesi terbesar yang berkaitan dengan informatika, yaitu Association of Computing Machinery (ACM) dan Institute of Electrical and Electronics Engineers (IEEE) telah merumuskan kurikulum yang dapat menjadi acuan untuk mempelajari informatika. ACM merupakan organisasi di bidang informatika yang paling tua, sedangkan IEEE merupakan

organisasi profesi di bidang elektro yang kemudian juga mencakup bidang informatika. Kurikulum di tingkat Pendidikan Dasar dan Menengah telah dibuat sejak tahun 2000, sedangkan kurikulum untuk tingkat perguruan tinggi telah dibuat sejak tahun 1990. Kurikulum tersebut telah dirilis oleh kedua organisasi tadi dan telah diperbarui beberapa kali untuk menyesuaikan dengan perkembangan terkini di dunia saat ini.

Pada pendidikan dasar dan menengah, informatika masih merupakan satuan kemasan keilmuan. Di tingkat sekolah dasar (SD), area pengetahuan yang dikenalkan dititikberatkan pada kemampuan berpikir komputasional. Pada tingkat sekolah menengah pertama (SMP), area pengetahuan yang diajarkan ditambah dengan dikenalkannya area teknologi informasi dan komunikasi (TIK) serta penerapannya yang menunjang kehidupan sehari-hari. Pada konteks kegiatan belajar-mengajar di sekolah, penerapan tersebut meliputi penerapan TIK untuk membantu murid belajar, berkolaborasi, berkomunikasi, serta mencari informasi. Pada tingkat Sekolah Menengah Atas (SMA), murid diajak berkreasi dan berinovasi untuk menghasilkan karya-karya digital untuk menjawab suatu tantangan dalam kehidupan.

Pada tingkat Perguruan Tinggi, setiap pilar pengetahuan dikemas menjadi satu program studi spesifik. Pada saat ini, terdapat lima jenis program studi yang didefinisikan baik oleh ACM dan IEEE dan diadopsi oleh Indonesian *Chapter of Association for Information Systems* (Aisindo). Program studi, baik di Indonesia maupun di negara lain, mengacu pada standar kurikulum tersebut untuk memastikan lulusannya memiliki kompetensi yang memiliki kemampuan kompetitif yang standar. Artinya, kompetensi yang dimiliki oleh lulusan diharapkan dapat diterima dan mampu bersaing di dunia internasional. Perguruan tinggi kemudian dapat mengadaptasi atau menyesuaikan kurikulum mereka untuk menciptakan karakter atau kekhasan lulusan setiap perguruan tinggi.

Berdasarkan Computing Curricula tahun 2005, terdapat lima disiplin ilmu komputasi yang paling besar, yaitu seperti berikut.

1. *Computer Engineering* atau Teknik Komputer yang mempelajari perancangan dan pembuatan komputer dan sistem berbasis komputer. Hal ini dapat mencakup perangkat keras (hardware), perangkat lunak (software), serta komunikasi dan interaksi antara perangkat keras dan perangkat lunak.
2. *Computer Science* atau Ilmu Komputer yang mempelajari teori dan algoritma untuk (a) pengembangan robotika, computer vision, sistem cerdas, bioinformatika, dan area lainnya; (b) perancangan dan implementasi perangkat lunak, seperti pembuatan program

komputer; (c) pembuatan teknologi untuk memanfaatkan komputer seperti World Wide Web, robot cerdas, dan analisis DNA manusia; dan (d) pengembangan metode-metode penyelesaian masalah komputasi secara efektif dan efisien seperti: metode terbaik untuk menyimpan informasi dalam suatu basis data, pengiriman data dalam sistem jaringan komputer, atau menampilkan gambar-gambar kompleks.

3. *Information System* atau Sistem Informasi yang mempelajari pada integrasi solusi-solusi teknologi informasi dengan tujuan-tujuan bisnis dan organisasi untuk mencapai proses yang lebih efektif dan efisien.
4. *Information Technology* atau Teknologi Informasi yang mempelajari solusi-solusi teknologi komputer bagi organisasi bisnis, pemerintah, Kesehatan, sekolah, dan organisasi lainnya.
5. *Software Engineering* atau Rekayasa Perangkat Lunak yang mempelajari pengembangan dan perawatan sistem perangkat lunak agar dapat bekerja dengan lebih reliabel, efisien, skalabel, dan terjangkau secara biaya untuk dikembangkan dan dirawat serta memenuhi kebutuhan penggunanya.

Kelima disiplin tersebut memiliki perbedaan dan kesamaan yang dapat divisualisasikan dengan mudah pada gambar yang tersaji di bawah. Pada gambar tersebut, sumbu horizontal mewakili seberapa teoretis atau aplikatif suatu bidang ilmu. Makin ke kanan, makin aplikatif bidang ilmu tersebut. Sumbu vertikal menggambarkan bidang-bidang yang meliputi perangkat keras komputer dan arsitektur, infrastruktur sistem, metode dan teknologi perangkat lunak, teknologi aplikasi, serta permasalahan organisasi dan sistem organisasi. Makin ke atas, makin "dekat" bidang tersebut dengan manusia dan organisasi.

Dari gambar tersebut, sebagai contoh, disiplin Teknik Komputer akan mencakup hal yang bersifat sangat teoretis hingga sangat aplikatif, tetapi hanya terbatas pada perangkat keras komputer dan arsitektur, sebagian infrastruktur sistem, dan sebagian kecil dari metode dan teknologi perangkat lunak. Di sisi lain, disiplin Teknologi Informasi sangat menitikberatkan pada hal-hal yang lebih aplikatif atau teknis mulai dari infrastruktur sistem hingga masalah organisasi dan sistem informasi.

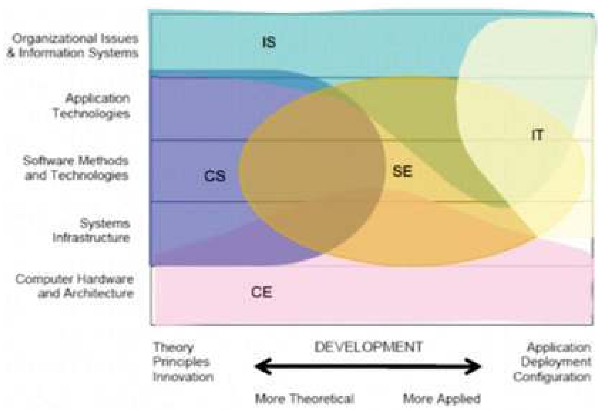
Kurikulum tersebut merupakan sebuah dokumen hidup yang terus diperbarui oleh ACM dan IEEE berdasarkan masukan-masukan dari berbagai pemangku kepentingan, yang termasuk berbagai perusahaan bereputasi di dunia. Versi pertama terbit pada tahun 1990, kemudian mengalami berbagai perubahan hingga terbitlah versi Computing Curricula 2003. Versi terbaru, yang pada saat buku ini ditulis masih dalam versi draf, telah mengalami 44 kali proses perbaikan pada tanggal 15 November 2020!

Ayo Berdiskusi

Ayo, Pilih Program Studimu di Perguruan Tinggi

Ayo, pikirkan bidang studi apa yang kalian cita-citakan. Untuk setiap poin di bawah ini, pikirkanlah dengan baik berdasarkan paparan di atas. Kalian juga bisa melakukan pencarian di internet atau membaca buku di perpustakaan untuk menjadi bahan berpikir. Kemudian, tuliskan hasil pemikiran kalian pada buku catatan.

1. Apakah kalian berminat untuk studi lanjut di bidang informatika? Mengapa?
2. Jika berminat, program studi apa yang akan kalian pilih? Jelaskan alasannya.
3. Apabila tidak berminat, menurut kalian, adakah peran dari informatika dalam bidang yang kalian minat? Jelaskan seperti apa peran informatika menurut pemahaman kalian.



Gambar 8.11 Lima Bidang Studi Komputing (Informatika)
Sumber: Dokumen Kemendikbud, 2021 diadaptasi dari ACM.org

3. Sertifikasi di Bidang Informatika

Kompetensi di bidang informatika tidak hanya dapat diperoleh melalui jalur pendidikan tinggi yang disahkan dalam bentuk ijazah. Belajar informatika bisa dari sumber apa pun. Sangat umum ditemukan seorang pengembang perangkat lunak yang belajar pemrograman secara otodidak. Pada saat ini, di Indonesia juga telah banyak kegiatan pelatihan, seminar, pelatihan, atau workshop yang dapat diikuti oleh masyarakat umum termasuk siswa sekolah menengah.

Belajar dan berlatih pun dapat dilakukan dengan basis komunitas. Di berbagai kota dan kabupaten di Indonesia, telah lahir berbagai komunitas di bidang informatika yang terbuka bagi siapa pun yang ingin belajar dan berbagi ilmu. Di internet, pun telah banyak tersedia sumber-sumber belajar yang dapat diakses, baik yang sifatnya gratis maupun berbayar dan memberikan sertifikat. Dengan kata lain, akses terhadap pendidikan informatika sangatlah luas dan memberi kesempatan bagi siapa saja untuk menguasai bidang ini. Tidak sedikit

siswa SMP atau SMA yang aktif di komunitas-komunitas dan telah membuat karya inovatif dan cemerlang walaupun ia belum secara formal mendapatkan materi informatika di jenjang pendidikannya. Hanya dua modal yang murid perlukan untuk belajar informatika, yaitu rasa ingin tahu dan semangat yang tinggi. Salah seorang ilmuwan terkemuka di bidang informatika, Prof. Jun Rekimoto, bahkan belajar memprogram tanpa komputer dengan menulis kode program di kertas dan membayangkan kode tersebut berjalan.

Walaupun demikian, kompetensi yang diperoleh seseorang perlu diakui. Salah satu bentuk pengakuan tersebut berupa sertifikasi. Sertifikasi ini merupakan suatu dokumen pengakuan yang diperoleh seseorang setelah dianggap memenuhi standar kompetensi yang telah ditetapkan oleh badan, organisasi, atau perusahaan yang memiliki keahlian di bidang tersebut. Biasanya, sertifikasi ini diberikan setelah seseorang lulus dari ujian yang diberikan. Seseorang yang telah memiliki sertifikat akan dapat menggunakan sertifikat tersebut saat bekerja. Hal ini dapat memberi kepercayaan dari tempat bekerja terhadap keahlian yang dimiliki oleh pekerjanya. Berdasarkan kepercayaan tersebut, upah yang diberikan kepada mereka yang memiliki sertifikasi seringkali lebih tinggi dibanding mereka yang belum memiliki sertifikasi.

Sertifikasi dapat diperoleh dari produsen perangkat keras maupun perangkat lunak, Lembaga Profesional, Asosiasi Profesi, atau Pemerintah. Sertifikasi dapat terikat pada suatu produk (misal sertifikasi penggunaan perangkat lunak perkantoran) ataupun pada suatu kemampuan (missal sertifikasi sebagai seorang data *scientist*). Lembaga Sertifikasi fokus kepada definisi kompetensi dan pelaksanaan ujian, tidak mempedulikan proses belajar sebelumnya. Artinya, seseorang tanpa belajar formal juga dapat mengikuti sertifikasi. Beberapa perusahaan bereputasi yang menyediakan sertifikasi yang diakui secara internasional antara lain seperti berikut.

1. Microsoft: Menawarkan sertifikasi mulai dari sertifikasi pemakaian produk seperti Microsoft Office, .Net, SQL Server, dan Azure serta sertifikasi terkait keprofesian seperti data scientist dan data engineer. Microsoft juga memberikan gelar Microsoft Most Valuable Professional bagi para profesional yang unggul dan berpengaruh di bidangnya.
2. Google: Menawarkan sertifikasi penggunaan produk seperti Google Suite dan juga sertifikasi terkait keprofesian seperti Android dan Google Cloud Platform. Google memberikan gelar Google Developer Expert bagi para profesional yang unggul dan berpengaruh di bidangnya. Google memiliki beberapa mitra di Indonesia dan menawarkan berbagai pelatihan gratis sebelum mengambil sertifikasi yang disediakan, misalnya lewat Google Developers Kejar dan beasiswa.

3. Oracle: Menawarkan sertifikasi terkait produknya seperti Java, OCJP, dan SCJP.
4. CISCO: Menawarkan pelatihan terkait jaringan seperti CCNA.

Selain itu, Kementerian Komunikasi dan Informatika Republik Indonesia juga menyediakan program pelatihan dalam rangka sertifikasi, misalnya program Pelatihan dan Sertifikasi Talenta Digital yang bekerja sama dengan perusahaan-perusahaan yang memberikan sertifikasi, dan juga bekerja sama dengan Perguruan Tinggi

Suatu sertifikasi yang kompleks dapat terdiri atas beberapa sertifikasi yang lebih dasar sehingga membentuk semacam jalur sertifikasi (certification track). Misalnya, untuk mendapatkan sertifikasi keahlian menggunakan aplikasi paket perkantoran, kalian harus terlebih dahulu mendapatkan sertifikasi keahlian di setiap aplikasi di paket tersebut, misalnya aplikasi pengolah kata, pengolah lembar kerja dan presentasi. Kalian dapat mengecek jalur sertifikasi yang ditawarkan di web setiap penyedia sertifikasi.

Bentuk ujian sertifikasi bisa berupa soal-soal teori ataupun praktik langsung. Misalnya, pada sertifikasi teknologi tingkat dasar, ujian sertifikasi dapat berupa soal-soal pilihan ganda untuk menguji pemahaman peserta ujian. Akan tetapi, sertifikasi juga dapat berupa soal-soal praktik untuk menguji keterampilan peserta dalam melaksanakan suatu tugas yang terkait dengan kompetensi yang diujikan. Sertifikasi ada yang dapat dilakukan secara online dengan pengawasan tertentu, tetapi ada yang mengharuskan peserta untuk hadir di suatu lokasi untuk diawasi secara langsung pada saat ujian.

Dua hal yang biasanya menjadi kekhawatiran dalam mengambil sertifikasi adalah tekanan untuk lulus sertifikasi dan masalah biaya. Beberapa sertifikasi yang berbayar tidak akan mengembalikan biaya pendaftaran terlepas kalian lulus atau tidak dari ujian sertifikasi tersebut. Oleh karena itu, yang paling penting adalah menguasai bidang yang akan disertifikasi dengan banyak belajar dan latihan sebelumnya. Selain itu, ada baiknya kalian mencoba mempelajari bagaimana bentuk ujian sertifikasi yang diberikan serta mencoba latihan sertifikasi yang biasanya diberikan oleh lembaga tersebut. Biasanya, contoh-contoh untuk mencoba disediakan dan dapat diakses bebas biaya

Dari sisi biaya, di Indonesia, cukup banyak akses pelatihan untuk sertifikasi yang dapat diikuti dengan gratis atau berbiaya murah. Selain itu, beberapa perusahaan juga memberi kesempatan berupa beasiswa pelatihan atau sertifikasi yang dapat diambil oleh masyarakat umum, termasuk siswa SMA. Akan tetapi, untuk mendapatkan informasi tersebut, perlu ada tindakan aktif dan inisiatif dari murid untuk mencari informasi. Cara yang

paling mudah ialah dengan rajin mengikuti akun-akun berita, perusahaan, komunitas, atau lembaga terkait di media sosial. Dengan cara demikian, selain menjalin komunikasi dengan temannya di media sosial, kalian juga dapat terpapar dengan informasi dan akses yang bermanfaat bagi masa depan karier dan pengembangan potensi diri kalian dengan cara yang menyenangkan.

Ayo Berdiskusi

Sertifikasi apa yang kalian minati?

Di waktu senggang, kalian dapat belajar dan mengambil sertifikasi yang terkait dengan minat dan pelajaran yang telah kalian dapatkan, misalnya untuk pemakaian paket aplikasi perkantoran. Sertifikasi ini akan berguna dalam hidup kalian kelak, terutama saat masuk ke dunia kerja, apa pun bidang studi yang kalian pilih karena kemampuan menggunakan komputer untuk penyelesaian persoalan sudah menjadi salah satu kompetensi umum. Mengikuti sertifikasi adalah tabungan kalian di masa depan. Sekarang, cari dan pelajarilah informasi mengenai sertifikasi-sertifikasi di atas serta ikuti pelatihan persiapan yang tersedia. Manfaatkan waktu luang dengan menabung untuk masa depan. Setelah selesai, jawablah pertanyaan berikut di buku tugas kalian.

1. Sertifikasi apa yang akan kalian ambil? Mengapa?
2. Apa peran sertifikasi tersebut pada kehidupan kalian sekarang dan di masa depan? Jelaskan.

Ayo Kita Renungkan

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja. Jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada materi ini, kalian telah memahami bagaimana informatika dapat berperan di berbagai bidang, bagaimana cara memperoleh pengakuan kompetensi di bidang informatika, serta aspek ekonomi dan hukum yang terkait dengan kehidupan sehari-hari. Dari materi tersebut, apakah kalian telah memahami alasan perlunya memiliki pengetahuan di bidang informatika, walaupun cita-cita kalian mungkin berada di bidang lain?
2. Setelah mengetahui tentang program studi di bidang informatika, apakah kalian tertarik untuk menempuh studi lanjut di Informatika? Jika ya, di bidang apa?

3. Setelah mengetahui tentang sertifikasi dan karier di bidang informatika, kompetensi informatika apa yang ingin kalian kuasai? Sertifikasi apa yang akan kalian ambil dalam waktu dekat?
4. Setelah mengetahui aspek dan dampak ekonomi dari informatika, apakah kalian tertarik untuk mengembangkan bisnis yang memanfaatkan informatika?
5. Setelah mengetahui aspek hukum dan hak kekayaan intelektual di bidang informatika, apakah kalian akan menjadi makin menghargai karya-karya di bidang informatika?

Apakah kalian akan mulai menggunakan perangkat lunak, buku, musik, atau file digital lain dengan lisensi yang benar, dengan menghargai hak para penciptanya?

Ingin Tahu Lebih?

Jika kalian ingin mengetahui lebih banyak tentang materi pada bab ini, kalian dapat merujuk beberapa situs berikut:

Bahan bacaan tentang sejarah komputer:

1. en.wikipedia.org/wiki/History_of_computing
2. cs.virginia.edu/~robins/The_Origins_of_Computing.pdf
3. en.wikipedia.org/wiki/List_of_pioneers_in_computer_science
4. en.wikipedia.org/wiki/List_of_computing_people
5. livescience.com/20718-computer-history.html
6. vig.prenhall.com/samplechapter/0130898155.pdf
7. egov.uok.edu.in/elearningug/tutorials/5786_2_2016_161115130838.pdf

Bahan bacaan tentang aspek ekonomi dan hukum produk informatika:

1. kemenparekraf.go.id/asset_admin/assets/uploads/media/pdf/media_1589834401_Infografis_Sebaran_Pelaku_Ekonomi_Kreatif.pdf
2. perpustakaan.bappenas.go.id/lontar/file?file=digital/192395-[_Konten_]-konten%20E2521.pdf
3. www8.cao.go.jp/cstp/english/society5_0/index.html
4. en.m.wikipedia.org/wiki/List_of_the_largest_software_companies
5. en.m.wikipedia.org/wiki/List_of_largest_Internet_companies
6. en.m.wikipedia.org/wiki/List_of_largest_technology_companies_by_revenue
7. www.fsf.org/

Bahan bacaan tentang informatika untuk masa depan:

1. Sachs JD. From millennium development goals to sustainable development goals. The Lancet. 2012 Jun 9;379(9832):2206-11.

2. sdg2030indonesia.org
3. sdgs.un.org/goals
4. weforum.org/agenda/2016/12/by-2030-this-is-what-computers-will-do
5. weforum.org/agenda/2016/09/7-innovations-that-could-shape-the-future-of-computing

Bahan bacaan tentang karier di bidang informatika:

1. Denning, P. J. (2001). The profession of IT: who are we?. Communications of the ACM, 44(2), 15–19. doi:10.1145/359205.359239
2. CODE.ORG (2020) CODE.org :"..... 398,857 Open computing jobs nationwide; 71,226 Computer science students graduated into the workforce last year, fromhttps://code.org/promote

Bahan bacaan tentang program studi di bidang informatika:

1. computer.org/volunteering/boards-and-committees/professional-educational-activities/curricula
2. ieeecs-media.computer.org/assets/pdf/CC2005-March06Final.pdf
3. ieeecs-media.computer.org/assets/pdf/se2014.pdf
4. ieeecs-media.computer.org/assets/pdf/CS2013-final-report.pdf
5. ieeecs-media.computer.org/assets/pdf/ce2016-final-report.pdf
6. acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf

Bahan bacaan tentang sertifikasi di bidang informatika:

1. docs.microsoft.com/en-us/learn/certifications/browse/
2. developers.google.com/certification
3. skillshop.withgoogle.com/
4. education.oracle.com/oracle-certified-professional-java-se-8-programmer/trackp_357
5. www.cisco.com/c/m/en_sg/partners/cisco-networking-academy/index.html

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas X
Penulis : Dean Apriana Ramadhan
ISBN : 978-602-244-506-7

Bab 9
Praktika Lintas Bidang Informatika



Sumber: Dokumen Kemendikbud, 2021

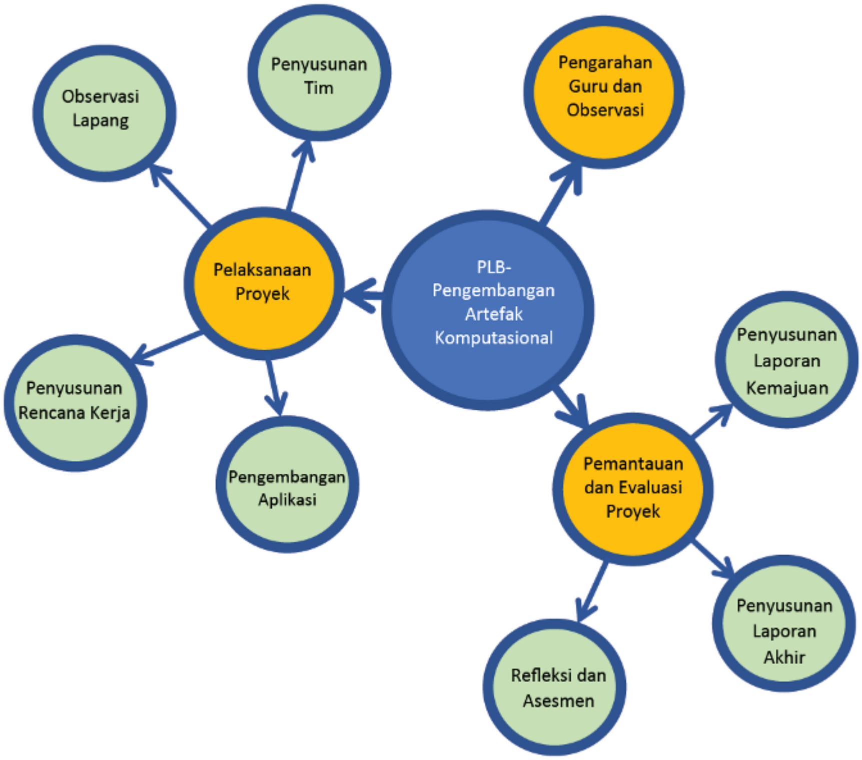
Tujuan Pembelajaran

Setelah mempelajari bab ini, kalian mampu berkolaborasi, mengenali, dan mendefinisikan persoalan yang penyelesaiannya dapat didukung dengan sistem komputasi, mengembangkan abstraksi untuk memodelkan masalah, mengembangkan artefak komputasional dengan membuat desain program sederhana, dan mempresentasikan hasil artefak tersebut.

Pertanyaan Pemantik

Aktivitas keseharian apa saja yang pernah kalian lakukan dengan bantuan teknologi informasi? Aplikasi apa yang pernah kalian gunakan untuk membantu menyelesaikan pekerjaan kalian? Permasalahan apa saja yang terdapat di lingkungan sekitar kalian, yang sebenarnya dapat diselesaikan dengan memanfaatkan teknologi informasi?

Peta Konsep



Gambar 9.1 Peta Konsep Praktik Lintas Bidang

Apersepsi

Pernahkah kalian berbelanja *online* di *e-commerce*? Bagaimana pengalaman kalian saat berbelanja *online*? Momen berbelanja merupakan salah satu bentuk revolusi penggunaan teknologi informasi dalam penyelesaian masalah sehari-hari. Sebelum internet booming, berbelanja hanya dilakukan secara tatap muka. Saat ini, *e-commerce* hadir sebagai solusi berbelanja dari gawai. Klik, pilih barang, pilih pengiriman, dan bayar. Kita cukup menunggu barang datang ke rumah. Sebelumnya, belanja tidak semudah ini.

Kata Kunci

Dokumentasi, observasi, kelompok, manajemen proyek, pengembangan aplikasi, desain aplikasi

Pengalaman berbelanja *online* merupakan salah satu bentuk nyata pemanfaatan teknologi informasi dalam kehidupan keseharian kita. Tentu, dalam proses pengembangan *e-commerce* agar bisa sebesar sekarang, banyak langkah yang perlu dilalui. Langkah awal pasti memerlukan analisis masalah dan mencari berbagai macam alternatif solusi. Proses dilanjutkan dengan mengembangkan aplikasi, lalu melakukan pengujian ke masyarakat. Proses mengembangkan sebuah aplikasi sebenarnya merupakan salah satu kegiatan yang memerlukan manajemen proyek.

Manajemen proyek dalam Kamus Besar Bahasa Indonesia diartikan sebagai telaah atau kumpulan pengetahuan tentang cara-cara menginisiasi, menjalankan, memonitor, dan mengevaluasi proyek. Pada bagian ini, kalian akan belajar untuk membangun sebuah desain aplikasi untuk menyelesaikan sebuah permasalahan yang ada di sekitar lingkungan kalian secara berkelompok. Aktivitas ini merupakan salah satu bentuk praktik lintas bidang dalam mata pelajaran Informatika.

Praktik Lintas Bidang (PLB) merupakan sebuah kegiatan pengembangan aplikasi yang dikerjakan dalam jangka waktu setengah semester (7 Minggu). Kegiatan ini diselenggarakan secara berkelompok dengan jumlah anggota di setiap kelompok 5-6 orang. Setiap kelompok akan diminta untuk melakukan observasi langsung ke lingkungan sekitar untuk mencari permasalahan yang akan dijadikan sebagai studi kasus PLB. Kalian akan berinteraksi dengan rekan-rekan sekelompok dan dengan masyarakat. Dalam bahasa lain, PLB adalah sebuah kegiatan yang dilakukan di luar jam sekolah dalam rangka implementasi ilmu Informatika yang telah didapat di sekolah. Kalian akan berlatih menyusun sebuah kegiatan proyek pengembangan aplikasi bersama dengan kelompok untuk menyelesaikan permasalahan yang kalian temui di masyarakat. Kalian akan mendapatkan pengarahan dari guru tentang PLB pada awal semester. Proyek ini akan melatih setiap anggota tim untuk mengelola setiap langkah pekerjaan. Produk dari PLB ini ialah aneka dokumentasi proyek dan tentu saja desain aplikasi yang dibangun.

A. Pengarahan Guru dan Observasi

Dalam proyek berjangka waktu agak lama ini, kalian akan bekerja dalam kelompok, dan juga akan mendapat bagian pekerjaan mandiri. Bekerja dalam kelompok perlu untuk membangun budaya kerja berkolaborasi yang kalian perlukan karena pada hakikatnya, suatu karya besar tidak mungkin dikerjakan sendiri. Sebuah karya besar adalah karya bersama sekumpulan orang, yang bahkan kelak, berasal dari berbagai latar belakang ilmu. Misalnya, karya besar

Patung Garuda Wishnu Kencana di Bali, atau Candi Borobudur di Jawa Tengah, Masjid Taj Mahal di India, itu bukan hanya karya seni yang dikagumi banyak orang, tetapi juga karya engineering karena pembangunannya membutuhkan perhitungan matang untuk dapat kokoh serta memerlukan perencanaan karena pengerjaannya harus bertahap dalam waktu lama. Seperti sebuah karya bangunan, sebuah program aplikasi juga dapat berskala besar. Coba kalian buka salah satu aplikasi untuk pemesanan transportasi online. Banyak sekali fiturnya, bukan? Tidak mungkin dikerjakan oleh satu individu saja. Aplikasi itu juga tidak dibuat begitu saja, tetapi setelah menggali kebutuhan dari masyarakat yang akhirnya memakainya.

Mari, kita latihan untuk membangun sebuah produk berupa program aplikasi yang akan berguna sebagai solusi permasalahan di sekitar kita, yang tidak terlalu besar tentunya, karena waktu kita yang terbatas. Kalian akan menyadari bahwa aplikasi komputer itu hanya sebagian dari solusi permasalahan yang ada.

1. Pengarahan Guru

Kalian akan mendapatkan penjelasan dari guru tentang pelaksanaan PLB ini. Sebagai permulaan, kalian wajib mencatat berbagai macam informasi yang guru sampaikan. Kalian dapat mencatat penjelasan guru dalam Buku Kerja Siswa menggunakan contoh format yang diberikan di bawah ini, yang tentu saja boleh kalian sesuaikan dengan gaya penulisan masing-masing. Bagi yang suka menggambar, boleh membuat komik sebagai sarana dokumentasi. Yang penting, elemen-elemen informasinya lengkap.

Hari / Tanggal kegiatan	
Narasumber	
Catatan	**Deskripsi kegiatan:**
	Yang perlu dipersiapkan:
	Timeline / rencana jadwal kegiatan PLB:

2. Observasi

Guru akan memberikan beberapa contoh kasus yang ada di sekitar lingkungan kalian. Salah satu contoh tema yang dapat diangkat ialah tentang konsep kota cerdas. Berikut ini contoh observasi kota cerdas.

KONSEP KOTA CERDAS

Menuju tahun 2045, beberapa kota di wilayah Indonesia mulai menerapkan konsep kota cerdas (*smart city*). Perkembangan teknologi informasi yang begitu pesat jelas memberikan dampak pada kehidupan bermasyarakat. Sebagai siswa yang telah belajar konsep-konsep Informatika, kalian akan melaksanakan sebuah kegiatan penerapan teknologi informasi dalam kehidupan bermasyarakat di lingkungan sekitar. Untuk menggali permasalahan yang ada, kalian dapat melakukan kunjungan dan wawancara terhadap masyarakat sekitar. Proyek-proyek smart city sederhana dapat meliputi konsep-konsep yang telah disusun oleh Kementerian Komunikasi dan Informatika, yaitu seperti berikut.

1. *Smart environment*: Menyiapkan kawasan wisata prioritas menjadi kawasan yang bersih, bebas sampah, dan tertib, tanpa meninggalkan unsur tradisionalnya.
2. *Smart economy*: Memastikan implementasi TIK dalam proses transaksi (*cashless*) berlangsung di kawasan wisata prioritas dan pemerintah daerah sekitarnya.
3. *Smart branding*: Membantu pemerintah daerah pada kawasan wisata prioritas dalam meningkatkan kunjungan wisata.
4. *Smart government*: Memastikan pemerintah daerah pada kawasan wisata prioritas menerapkan Sistem Pemerintahan Berbasis Elektronik (SPBE) secara berkualitas dalam upaya pelayanan publik yang baik.
5. *Smart society*: Memastikan masyarakat tujuan wisata prioritas dan kawasan sekitarnya memiliki kapasitas unggul dan mampu menjadi tuan rumah yang baik. .
6. *Smart living*: Mendorong situasi kawasan wisata prioritas yang kondusif dan nyaman bagi masyarakat dan wisatawan, melalui penyediaan transportasi, logistik yang tenteram, aman, dan ramah.

Sebagai studi kasus PLB kali ini, kalian akan diminta untuk dapat berkolaborasi dengan pengurus RT/RW yang akan ditentukan bersama guru, untuk menggali permasalahan tentang pendataan kependudukan dan aktivitas bulanan seperti:

1. Pendataan kelahiran
2. Pendataan kematian
3. Pendataan tamu
4. Penarikan iuran sampah
5. Penarikan iuran keamanan
6. Pendataan jumlah sampah yang dihasilkan warga
7. Pendataan luas wilayah RT

Kalian diminta untuk membangun sebuah sistem sederhana sebagai bagian dari solusi untuk menyelesaikan permasalahan-permasalahan tersebut.

Proyek ini akan dilakukan di lingkungan sekitar tempat kalian berada. Sebagai bagian dari observasi, kalian diharapkan dapat menggali keadaan lingkungan sekitar rumah dan mencoba berinteraksi dengan perangkat pengurus RT/RW. Kalian diharapkan dapat memformulasikan sebuah permasalahan yang dihadapi oleh pengurus RT dan RW seputar kependudukan. Selanjutnya, kalian akan membuat alternatif solusi dari permasalahan tersebut dan menyusun langkah-langkah terstruktur untuk menyelesaikannya. Sebagai produk dari proyek ini, kalian akan membuat sebuah program sederhana dalam bahasa pemrograman Python untuk menyelesaikan permasalahan yang ada di lingkungan RT/RW. Sengaja ditetapkan dalam bahasa pemrograman yang berbeda agar kalian memiliki pengalaman dan membuktikan bahwa kalian bisa mentransfer pola konsep pemrograman dalam bahasa C yang telah kalian kenal semester sebelumnya, ke bahasa Python.

Apa yang akan kalian dapatkan dalam PLB ini?

Proyek ini memiliki tujuan untuk:

1. Membangun sebuah sistem sederhana untuk membantu proses pengelolaan pekerjaan rutin di kepengurusan RT/RW setempat.
2. Memahami masalah serta mencari solusi untuk permasalahan tersebut.
3. Memberikan pengalaman berinteraksi secara langsung dengan masyarakat.
4. Menerapkan konsep informatika dalam kehidupan bermasyarakat.

B. Pelaksanaan Proyek

1. Penyusunan Tim/Kelompok

Setelah kalian mendengarkan arahan dan melakukan observasi awal, kalian akan diminta untuk menyusun tim/kelompok. Carilah 5-6 orang yang akan bekerja bersama kalian. Tuliskan dalam format berikut (kalian dapat menggunakan aplikasi pengolah data atau aplikasi pengolah kata untuk mencatat/mendokumentasikan hasil kerja kelompok kalian).

NAMA KELOMPOK				
No	Nama	Peran	No HP	Alamat email
1				
2				

2. Pembagian Peran

Setiap orang dalam kelompok akan mendapatkan perannya masing-masing. Peran yang dapat kalian pilih untuk setiap orang anggota kelompok seperti berikut.

No	Peran	Tugas/Pekerjaan
1	Project manager	• Menyusun jadwal kerja • Menyusun pembagian kerja kelompok • Membangun komunikasi dengan RT/RW • Memonitor jalannya proyek
2	System Analyst	• Melakukan analisis proses • Mendefinisikan kebutuhan pengurus RT/RW • Memodelkan kebutuhan aplikasi dalam bentuk prototipe
3	Programmer	• Menerjemahkan hasil analisis menjadi aplikasi • Menulis kode program
4	Tester	• Membuat skenario tes • Melakukan tes terhadap aplikasi yang dibuat • Mendokumentasikan hasil tes

Perhatikan bahwa peran-peran tersebut dibutuhkan dalam membangun solusi apa pun. Kalaupun solusinya bukan program komputer, analisis sistem tugasnya untuk mempelajari persoalan dan mencari solusi yang optimal. Programmer yang membangun produk terkait solusinya, tester yang akan menguji produk solusi. Tester biasanya ialah orang yang berbeda karena menguji program sendiri seringkali tidak bisa cermat, seperti kata pepatah 'gajah di pelupuk mata tidak kelihatan, semut di seberang lautan tampak jelas'. *Project manager* bertugas memastikan bahwa semua tugas dilaksanakan sesuai rencana.

Jika tugas sangat besar dan dikerjakan banyak orang, satu peran dapat dilakukan oleh lebih dari satu orang. Sebaliknya, apabila pekerjaan atau tugas sangat kecil, bisa saja satu orang menjalankan peran ganda, bahkan untuk tugas mandiri, semua peran terpaksa dilakukan oleh satu orang saja, yaitu diri kalian sendiri. Dalam hal ini, kalian harus tahu sedang berperan sebagai apa, agar tugas dapat dilaksanakan sebaik-baiknya.

3. Penyusunan Rencana Kerja

Setelah kalian mendapatkan kelompok dan peran, langkah awal yang perlu dilakukan ialah melakukan penyusunan rencana kerja. Penyusunan rencana kerja dapat menggunakan *gantt chart* sederhana dengan format sebagai berikut. *Gantt chart* adalah sebuah graph berarah yang merepresentasikan urutan paket-paket pekerjaan yang dilakukan. Sebuah proyek biasanya dibagi menjadi beberapa paket kerja, yang akan dikerjakan oleh satu orang atau lebih. Strategi pembagian anggota proyek ke paket kerja disesuaikan dengan peran anggota proyek, agar tujuan proyek dapat dicapai secara optimal.

No	Kegiatan	Bulan				Penanggung Jawab	Ket
		Minggu ke 1	Minggu ke 2	Minggu ke 3	Minggu ke 4		

Contoh penggunaannya adalah sebagai berikut:

No	Kegiatan	Bulan/Minggu																Penanggung Jawab	Keterangan
		1				2				3				4					
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
1	Observasi ke rumah Ketua RT.	■																Budi	tanggal 2 Januari 2021. bawa surat pengantar dari sekolah
2	Diskusi internal kelompok.		■															Ani	
3	Pembuatan laporan observasi lapang.			■	■													Joni	
4	Presentasi ke rumah Ketua RT.					■												Rustam	

Kolom KEGIATAN diisi dengan daftar aktivitas yang akan dikerjakan. Kolom BULAN dan MINGGU merupakan penanda kapan kegiatan tersebut harus dilaksanakan. Kolom PENANGGUNG JAWAB diisi dengan nama anggota kelompok yang merupakan penanggung jawab dari kegiatan tersebut. Berbagai catatan dapat diberikan di kolom KETERANGAN.

Secara garis besar, kegiatan PLB ini akan dilakukan berpusat kepada siswa dan berorientasi PBL (*project based learning*). Jadwal umum kegiatannya seperti dalam tabel berikut, detail jadwalnya akan ditentukan oleh guru.

No	Deskripsi Kegiatan	Luaran Hasil Siswa	Keterangan	Durasi
1	Pembagian Kelompok Kerja	Kelompok Kerja Siswa	Guru memimpin pembagian kelompok dan peran siswa dalam kelompok.	1 Hari / 2 JP
2	Penjelasan umum proyek	Rangkuman, Notulen pertemuan	Guru menjelaskan alur pengerjaan proyek dari Panduan Proyek yang disusunnya.	1 hari / 2 JP
3	Survei lapangan	Dokumen hasil survei	Siswa turun ke lapangan untuk menggali permasalahan	1 minggu

No	Deskripsi Kegiatan	Luaran Hasil Siswa	Keterangan	Durasi
4	Diskusi Internal Kelompok	Brainstorming, Dokumen tindak lanjut hasil survei	Kelompok siswa melakukan diskusi internal untuk membahas temuan-temuan pada saat survei lapangan	1 minggu
5	Presentasi awal proyek	Dokumen Presentasi Proyek	Setiap kelompok mempresentasikan proyek yang akan dikerjakan dan akan diberi masukan oleh guru	2 hari / 4 JP
6	Inisiasi proyek	Foto/Video	Kelompok siswa melakukan persiapan awal untuk turun ke lokasi observasi dan melakukan pemaparan ke masyarakat tentang proyek yang akan dilakukan	1 minggu
7	Pengerjaan proyek	Log kegiatan	Siswa melakukan pengerjaan proyek secara mandiri dipantau oleh Guru	1 minggu
8	Diskusi Guru - Siswa	Foto/Video, Produk	Guru mendampingi pengerjaan proyek yang dilakukan oleh siswa secara berkala	1 minggu
9	Diskusi Siswa - Masyarakat	Foto/Video, Produk	Siswa mempresentasikan hasil sementara dari proyek ke masyarakat.	1 minggu
10	Presentasi akhir ke klien	Foto/Video, Produk	Siswa mempresentasikan hasil proyek akhir ke klien.	2 hari / 4 JP
11	Presentasi akhir ke guru	Foto/Video, Produk, dokumen proyek	Siswa mempresentasikan hasil implementasi proyek di masyarakat kepada guru.	1 hari / 4 JP

4. Observasi Lapangan

Kegiatan ini merupakan kegiatan lapangan. Misalkan untuk contoh kasus Kota Cerdas, kalian akan diminta untuk berdiskusi dengan pengurus RT/RW di lingkungan tempat tinggal kalian. Topik yang akan diambil ialah permasalahan pendataan kependudukan atau pelayanan publik. Contoh masalah yang mungkin kalian digali ialah:

1. Apa saja permasalahan seputar kependudukan di RT/RW?
2. Bagaimana alur pelaksanaan atau prosedur dari sebuah pelayanan yang ada di lingkungan RT/RW?
3. Siapa saja yang terlibat dalam pelayanan di lingkungan RT / RW?

Hasil observasi lapangan dapat kalian tuliskan dalam format berikut.

Hari / Tanggal	
Lokasi	
Peserta Observasi Lapangan	1. 2. ...
Hasil Observasi Lapangan	
Nama pelayanan: .... **Alur pelayanan:** .... **Aktor yang terlibat dalam pelayanan:** .... **Permasalahan:**	
Rencana tindak lanjut	1. 2. ...

Setelah mendapatkan permasalahan di lapangan, kalian perlu melakukan diskusi dengan kelompok kalian untuk menentukan rencana tindak lanjut dari permasalahan yang sudah kalian temukan.

5. Perancangan dan Pengembangan Aplikasi

Peralatan, bahan, dan sarana yang dibutuhkan untuk mengerjakan proyek ialah:

1. Komputer/Laptop
2. Koneksi internet
3. Kertas HVS
4. Pulpen
5. Aplikasi pengolah kata (misal MS Word, Google Docs, dsb)
6. Dst, silakan kalian mengisi semua peralatan dan sarana yang mungkin akan kalian buat

Pengembangan desain aplikasi merupakan tujuan utama dalam proyek ini. Kalian dan kelompok kalian akan mengembangkan aplikasi secara mandiri. Kalian dapat melakukan browsing menggunakan internet untuk belajar secara mandiri mendesain alur aplikasi.

Petunjuk: kalian dapat melakukan observasi tentang sebuah aplikasi *e-commerce* dan mencoba berbelanja di kedua *platform* tersebut sambil mencatat alur ketika seorang pelanggan berbelanja di *e-commerce*. Untuk menggambarkan sebuah halaman, kalian dapat menggunakan aplikasi pengolah presentasi. Berdiskusilah dengan rekan kelompok kalian tentang apa saja yang perlu ada di halaman aplikasi yang kalian buat!



Gambar 9.2 Contoh Membuat Desain Aplikasi Dengan Menggunakan MS PowerPoint.

C. Monitoring dan Evaluasi Proyek

Selain mengembangkan aplikasi, salah satu luaran wajib dari proyek ialah rekaman aktivitas proyek yang berisi tentang daftar aktivitas individu dalam mengerjakan proyek. Luaran lainnya ialah kode program, desain prototipe, dan aplikasi. Selain dalam bentuk produk aplikasi, terdapat dokumen pendukung seperti poster/infografis serta dokumen yang berisi refleksi dan evaluasi.

Jurnal kelompok dapat menggunakan format sebagai berikut.

No	Nama Aktivitas	Hari/Tanggal/Jam	Pelaksana	Keterangan
1	Aktivitas A	Sabtu/13 Februari 2021/ 10.00	Siswa A	Membuat desain pertanyaan wawancara.
2				
3				
4				

Format rekaman aktivitas proyek secara sederhana sebagai berikut.

No	Nama Siswa:	Budi Sudarsono	
	Hari/Tanggal/Waktu	Aktivitas	Keterangan
1	Senin, 15 Februari 2021/ 08.00 – 10.00	Membuat laporan observasi.	Dikerjakan secara bersama menggunakan google docs di alamat https: https://docs.google.com/spreadsheets/d/1FxY3uAk.
2			
3			
4			

Setiap anggota kelompok wajib membuat rekaman aktivitas proyek (log activity) ini yang kemudian akan dikumpulkan ke guru pada saat penyusunan laporan kemajuan dan laporan akhir.

1. Penyusunan Laporan Kemajuan

Laporan kemajuan merupakan laporan singkat yang berisi tentang:

a. Deskripsi umum proyek
b. Temuan masalah di lapangan
c. Solusi yang diambil
d. Kendala pengembangan
e. Rencana tindak lanjut penyelesaian pengembangan aplikasi

Kalian dan kelompok kalian dapat membuat laporan singkat ini menggunakan aplikasi pembuat presentasi atau aplikasi pengolah kata.

2. Penyusunan Laporan Akhir

Laporan akhir merupakan laporan lengkap pengerjaan PLB. Laporan ini berisi hal-hal berikut.

a. Deskripsi umum proyek
b. Temuan masalah di lapangan
c. Solusi yang diambil
d. Proses desain dan pengembangan aplikasi
e. Demo aplikasi
f. Saran

Kalian dan kelompok kalian dapat membuat laporan singkat ini menggunakan aplikasi pembuat presentasi atau aplikasi pengolah kata. Selain laporan, pastikan kelompok kalian juga mengumpulkan desain prototipe dari aplikasi yang dibuat.

TANTANGAN

Apakah kelompok kalian dapat mengimplementasikan desain aplikasi kalian menjadi sebuah program utuh atau aplikasi? Kalian bebas mengimplementasikan aplikasi kalian dalam bahasa pemrograman apapun yang kalian kuasai.

3. Refleksi dan Evaluasi Diri

Selain guru yang memberikan evaluasi, kalian juga diberi kesempatan untuk menyusun evaluasi atau refleksi diri. Refleksi diri bersifat individu dan juga kelompok. Setelah kalian mengisi refleksi diri dan kelompok, kalian dapat mengumpulkannya ke guru. Setidaknya, ada 3 aspek yang ditanyakan saat mengisi refleksi diri, yaitu deskripsi pengalaman yang kalian dapatkan, kendala yang kalian hadapi, serta tindak lanjut kalian terhadap pengalaman bekerja secara berkelompok maupun individu.

Form Evaluasi Diri Siswa **Nama Siswa: ....**
Selama mengerjakan proyek ini, saya mendapatkan: ....
Kendala yang saya hadapi: .... **Di masa yang akan datang, jika saya bekerja secara berkelompok, saya akan melakukan:** ....

Form Evaluasi Kelompok Siswa **Nama Kelompok: ....** **Anggota Kelompok: ....**
Selama mengerjakan proyek ini, kelompok saya mendapatkan pengalaman: .... **Kendala yang saya hadapi:** .... **Di masa yang akan datang, kelompok kerja saya saya akan melakukan perbaikan pada:** ....

Selain mengevaluasi diri kalian sendiri, kalian juga diberi kesempatan untuk memberikan penilaian kepada teman sekelompok. Kalian dapat menuangkan penilaian terhadap teman kelompok dalam format berikut. Penilaian itu akan menjadi bahan berharga untuk perbaikan diri teman kalian. Berikan kritik yang objektif dan saran membangun dengan sopan.

Form Evaluasi Individu terhadap Teman Satu Kelompok

No	Nama Siswa	Evaluasi Kualitatif
1	Siswa A	Siswa ini mampu menjadi pemimpin di kelompok ini. Mampu membangun kerja sama dan berkomunikasi dengan baik. Perlu ditingkatkan kemampuan menulisnya.
2	Siswa B	
3	Siswa C	
4	Siswa D	
5	Siswa E	

Refleksi Individual

1. Refleksi individual tentang "Proyekku"

Aspek	Refleksiku
Secara keseluruhan, apakah saya puas dan senang mengerjakan proyek ini? Mengapa?	
Apakah Proyekku Keren?	
Kerennya di mana?	
Pengalaman apa yang paling mengasyikkan dalam proyek ini?	
Pengalaman apa yang paling tidak menyenangkan dalam proyek ini?	
Apakah saya senang dengan peran saya? Andaikata boleh memilih peran lain, peran apa yang saya inginkan?	
Apakah saya puas dengan hasilnya?	
Adakah yang masih perlu saya perbaiki dan saya sempurnakan?	

2. Dari pengalaman mengerjakan proyek ini, tuliskan refleksi kalian tentang relasi proyek yang dikerjakan dengan konsep elemen pengetahuan mata pelajaran Informatika dan bidang lain. Jelaskan dengan ringkas.

Konsep	Yang perlu untuk mengerjakan proyek, yaitu untuk ....
BK	
TIK	
SK	
JKI	
AD	
AP	
DSI	
Bidang lain:.................	

Glosarium

alamat IP (*IP address*) sebuah identitas angka yang digunakan semua perangkat komputer agar saling berhubungan dalam jaringan internet

algoritma (*algorithm*) suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan dalam bentuk program komputer untuk menyelesaikan suatu permasalahan komputasi tertentu

antarmuka baris perintah (*command line interface*) tipe antarmuka melalui text-terminal

antarmuka grafis pengguna (*graphichal user interface*) antarmuka yang menggunakan menu grafis untuk memudahkan pengguna berinteraksi dengan komputer

antrean (*queue*) struktur data abstrak yang merupakan koleksi sekumpulan elemen, di mana penambahan elemen dilakukan di paling ujung yaitu pada "ekor" (*tail*), dan penghapusan elemen dilakukan pada titik pelayanan yang disebut "*head*. oleh karena itu antrean juga disebut *first in first out* (FIFO)

aplikasi pengolah bahan presentasi (*PowerPoint*) aplikasi yang digunakan menyiapkan presentasi dalam bentuk "*slides*" di mana pada umumnya teks disajikan dalam bentuk butir-butir ringkas dan gambar yang dapat disajikan per *slides*, atau sebagian demi sebagian sehingga presentasi menjadi animasi sederhana

aplikasi pengolah kata (*word processor*) aplikasi yang digunakan untuk mengolah dokumen yang sebagian besar berisi teks. versi awal pengolah kata hanya seperti mesin ketik, untuk mengolah huruf, kata dan teks; versi selanjutnya sudah menyediakan berbagai fitur yang bukan hanya untuk teks

aplikasi pengolah lembar kerja (*spreadsheet*) aplikasi yang digunakan untuk perkantoran dalam mengolah data dalam bentuk lembar kerja, yang membantu dalam melakukan perhitungan-perhitungan sederhana dan pengolahan data (perhitungan statistik, dan proses lain seperti pengurutan, filter, menghilangkan duplikasi dll)

aplikasi perkantoran (*office application*) suatu paket aplikasi yang semula digunakan untuk perkantoran (pekerjaan menyiapkan dokumen yang zaman sebelum komputer memakai mesin ketik), yang oleh microsoft disebut office application

badan program (*body program/program definition*) realisasi dari fungsi, kode program yang dituliskan di dalam sebuah fungsi

bahasa pemrograman (*programming language*) kumpulan perintah, instruksi, dan sintaks lain yang digunakan untuk membuat suatu program

binary code kode yang hanya terdiri atas angka 0 dan 1

blok (*block*) sekumpulan kalimat C yang ditulis di antara { dan }

busur (*edge*) garis yang menghubungkan 2 buah simpul dalam suatu graf

cakupan (*scope*) daerah program tempat suatu nama dikenal

camel case konvensi (kesepakatan) penamaan yang menulis huruf pertama pada setiap kata (selain kata pertama) dengan huruf kapital contoh: camelCase

debugging Proses identifikasi dan perbaikan suatu kesalahan pada kode program

definisi (*definition*) memberitahukan sifat (*property*) objek dan sekaligus mengalokasikan memori untuk objek

deklarasi (*declaration*) memperkenalkan suatu entitas sebagai bagian dari suatu lingkup program, yang meliputi memberikan nama dan sifat dari objek contoh: deklarasi variabel

deklarasi global (*global declaration*) deklarasi yang berlaku dalam satu unit translasi (*file*)

deklarasi lokal (*local declaration*) deklarasi yang hanya berlaku dalam blok tempat deklarasi

diagram alir (*flowchart*) sebuah bagan atau diagram dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail serta hubungan antar proses

ekspresi (*expression*) komponen bahasa pemrograman yang mendefinisikan suatu proses perhitungan/komputasi

executable code berkas yang saat dimuat ke dalam memori dapat langsung dieksekusi oleh komputer

firewall perangkat lunak yang digunakan untuk menjaga keamanan jaringan pribadi. Firewall memblokir akses tidak sah ke atau dari jaringan pribadi dan sering digunakan untuk mencegah pengguna web yang tidak sah atau perangkat lunak terlarang mendapatkan akses ke jaringan pribadi yang terhubung ke internet

fitur penggabung surat (*mail merge*) fitur yang ada pada Microsoft Word yang berguna untuk pembuatan surat dan dokumen lain yang isinya sama, tetapi identitas penerimanya saja yang berbeda

General Packet Radio Service (GPRS) suatu teknologi yang memungkinkan pengiriman dan penerimaan data lebih cepat dibandingkan dengan penggunaan teknologi *Circuit Switch Data* atau CSD

graf (*graph*) suatu struktur dari sekumpulan objek di mana beberapa pasangan objek memiliki hubungan atau keterkaitan tertentu

graf berarah (*directed graph*) suatu graf di mana setiap sisi yang menghubungkan simpul-simpulnya memiliki arah tertentu

graf tidak berarah (*undirected graph*) suatu graf di mana semua sisi yang menghubungkan simpul-simpulnya tidak memiliki arah tertentu

header file berisi deklarasi fungsi c dan definisi makro untuk dibagikan antara beberapa file sumber

identifier (*identifier*) sekumpulan karakter yang digunakan untuk mengidentifikasi elemen-elemen dari program, seperti variabel dan fungsi

Ilmu Komputer (*Computer Science*) studi tentang komputer dan proses algoritmik, termasuk prinsip-prinsipnya, desain perangkat keras dan perangkat lunaknya, implementasinya, dan dampaknya terhadap masyarakat

Informatika (*Informatics*) ilmu yang mempelajari penggunaan komputer untuk mengatur dan menganalisis data yang berukuran besar

inisialisasi (*initialization*) memberikan nilai (operasi penugasan) awal pada suatu variabel

inklusif (*inclusive*) dalam konteks Matematika dan Informatika, inklusif berarti 'termasuk'. keterangan 1 sampai 100 (inklusif) artinya kalian dapat memilih bilangan 1, 100, dan semua bilangan di antara 1 dan 100

integer tipe data yang merepresentasikan bilangan bilangan bulat

internet (*internet*) jaringan komputer global yang koneksinya menggunakan protokol bersama (dalam hal struktur dan bahasa untuk permintaan file antara klien dan server) untuk berkomunikasi

jaringan komputer (*computer network*) koleksi dari dua atau lebih komputer yang dihubungkan bersama-sama untuk tujuan berbagi informasi, dan sumber daya, antara satu sama lain

kabel UTP (*UTP cable*) salah satu perangkat keras komputer berupa kabel yang digunakan sebagai penghubung atau transmisi data pada sebuah jaringan

kasus uji suatu kumpulan kondisi atau nilai yang dimasukkan ke dalam program oleh penguji untuk menentukan apakah sistem yang diuji memenuhi spesifikasi atau berjalan dengan benar

keluaran (*output*) hasil yang diperoleh dari suatu program yang berjalan yang dikirimkan ke luar dari program, misalnya kepada manusia atau program lainnya

keputusan bersarang (*nested if*) Struktur kontrol keputusan yang berada di dalam struktur kontrol keputusan yang lain

kesalahan sintaks (*syntax error*) kesalahan pada pemrograman yang terjadi ketika kode yang ditulis melanggar aturan sintaks dari suatu bahasa pemrograman; kesalahan ini dapat dideteksi oleh kompilator

kesalahan semantik (*semantic error*) Kesalahan pada pemrograman karena kesalahpahaman pada makna atau efek dari suatu elemen pada bahasa pemrograman

klien komputer (*computer client*) penerima akhir penerima atau pemohon layanan dalam jenis sistem model klien/server

kode sumber (*source code*) bentuk program yang diberikan kepada kompilator untuk dikonversi menjadi object code

konstanta (*constant*) suatu nilai yang tidak berubah

masukan (*input*) data yang dimasukkan ke dalam program untuk diproses lebih lanjut

memori (*memory*) media penyimpanan data pada komputer

mnemonic peranti seperti pola huruf, ide, atau asosiasi yang membantu dalam mengingat sesuatu

operasi biner Suatu operasi yang melibatkan dua buah operan.

operasi uner Operasi yang hanya melibatkan satu buah operan.

operator Suatu fungsi yang dilakukan pada suatu operan.

operator aritmatika Jenis operator yang digunakan untuk melambangkan suatu operasi aritmatika, seperti tambah, kurang, kali, dan bagi.

operator decrement Operator yang digunakan untuk mengurangi nilai dari suatu variabel, misalnya '--'.

operator increment Operator yang digunakan untuk menambah nilai dari suatu variabel, misalnya ++.

operator logika Operator yang digunakan untuk menghasilkan nilai benar atau salah (*boolean*).

operator majemuk (*compound operator*) Bentuk singkat penulisan operator penugasan, misalnya x = x + 1 ditulis x += 1.

object code berkas yang dihasilkan oleh kompilator

peladen (*server*) komputer atau program komputer yang didedikasikan untuk serangkaian tugas tertentu yang menyediakan layanan ke komputer atau program lain di jaringan

pemacu peranti (*device driver*) komponen perangkat lunak yang mengizinkan sebuah sistem komputer untuk berkomunikasi dengan sebuah perangkat keras

pembajakan (*piracy*) penyalinan, distribusi, atau penggunaan perangkat lunak ilegal

pencarian (*searching*) suatu jenis permasalahan pada komputasi yang mencari suatu objek yang memenuhi kriteria tertentu dari sekumpulan objek

pencarian mendalam (*depth first search*) algoritma untuk menelusuri graf dengan cara mengunjungi simpul anak sebelum mengunjungi simpul saudara, sehingga penelusuran dilakukan mengikuti alur secara mendalam pada suatu jalur terlebih dahulu sebelum melebar

pencarian menyebar (*breadth first search*) algoritma untuk menelusuri graf dengan cara mengunjungi simpul saudara sebelum mengunjungi simpul anak sehingga penelusuran dilakukan mengikuti alur secara melebar pada suatu jalur terlebih dahulu sebelum mendalam

penelusuran (*code tracing*) menginterpretasikan hasil dari setiap baris kode dan melacak dengan manual efek dari setiap pernyataan

penelusuran graf (*graph traversal*) proses penelusuran simpul dalam graf berdasar sisi yang menghubungkan simpul

pengguna (*user*) orang yang menggunakan komputer atau layanan jaringan

pengurutan (*sorting*) suatu jenis permasalahan pada komputasi untuk mengurutkan suatu himpunan objek berdasarkan kriteria tertentu

penugasan (*assignment*) suatu pernyataan yang memberikan sebuah nilai pada suatu variabel

perangkat keras (*hardware*) komponen fisik yang menyusun sistem komputasi, komputer, atau perangkat komputasi

perangkat lunak (*software*) program yang berjalan di atas sistem komputasi, komputer, atau perangkat komputasi lainnya

perangkat penghubung (*hub*) perangkat keras yang menyampaikan data komunikasi. Sebuah hub mengirimkan paket data (*frame*) ke semua perangkat di jaringan, terlepas dari alamat MAC yang terdapat dalam paket data tersebut

perangkat peralih (*switch*) perangkat berkecepatan tinggi yang menerima paket data yang masuk dan mengarahkannya kembali ke tujuannya di jaringan area lokal (LAN)

pernyataan (*statement*) unit bahasa pemrograman yang menyusun suatu program

perulangan bersarang (*nested loop*) Struktur kontrol perulangan yang berada di dalam struktur kontrol perulangan yang lain.

portofolio (*portfolio*) beberapa koleksi gambar, dokumen, dan lain-lain yang merepresentasikan pekerjaan orang tersebut

postfix increment/prefix decrement Operator increment atau decrement yang diletakkan sebelum variabel. Operator prefiks akan meningkatkan nilai suatu variabel, baru mengembalikan hasilnya untuk digunakan.

prefix increment/prefix decrement Operator increment atau decrement yang diletakkan sebelum variabel. Operator prefiks akan menambah atau mengurangi nilai suatu variabel, baru mengembalikan hasilnya untuk digunakan

program (*program*) sekumpulan pernyataan yang dapat dieksekusi oleh komputer untuk menghasilkan perilaku yang diinginkan dari komputer

prototipe (*prototype*) dalam bahasa C, artinya deklarasi fungsi, menyatakan nama, tipe return value, nama dan tipe parameter formal (argumen)

pseudokode (*pseudocode*) deskripsi program informal yang tidak mengandung sintaks kode atau pertimbangan teknologi yang mendasari

rekursi (*recursion*) suatu bentuk pendefinisian sebuah struktur yang mengandung dirinya sendiri. fungsi/barisan rekursif adalah fungsi/barisan di mana nilainya tersebut ditentukan/tergantung dari nilai fungsi/barisan itu sendiri, pada urutan nilai-nilai sebelumnya

router perangkat yang menganalisis konten paket data yang dikirimkan dalam jaringan atau ke jaringan lain

semantik (*semantic*) bagian dari definisi bahasa pemrograman yang terkait dengan makna dari sesuatu teks yang disusun mengikuti aturan penulisan sintaks dari suatu bahasa pemrograman

sertifikasi (*certification*) Pengakuan formal mengenai pencapaian atau kompetensi seseorang

Sertifikat SSL SSL merupakan singkatan dari Secure Sockets Layer, suatu teknologi keamanan standar global yang memungkinkan komunikasi terenkripsi antara peramban web dan server web

simpul (*node*) unit pokok (elemen) yang merepresentasikan suatu objek yang membentuk suatu graf

sintaks (*syntax*) aturan yang mendefinisikan cara menulis elemen bahasa pemrograman yang legal (harus dipatuhi oleh pemrograman) tanpa mempedulikan makna dari penulisan tersebut

sistem operasi (*operating system*) perangkat lunak sistem yang mengatur sumber daya dari perangkat keras dan perangkat lunak, serta sebagai daemon untuk program komputer

strategi (*strategy*) langkah terstruktur yang kalian lakukan untuk melakukan sesuatu. dalam hal ini, langkah tersebut kalian gunakan untuk melakukan penebakkan; strategi ini dapat ditulis juga sebagai sebuah algoritma

tersarang (*nested*) struktur kontrol yang menjadi bagian dari suatu struktur kontrol lain.

tumpukan (*stack*) struktur konseptual yang terdiri atas sekumpulan elemen homogen dan didasarkan pada prinsip terakhir masuk pertama keluar (*last in first out* - LIFO)

unit kontrol (*control unit*) unit kontrol (CU) menangani semua sinyal kontrol prosesor

unit logika aritmatika (*arithmetic logic unit*) komponen utama dari unit pengolah pusat dari sistem komputer yang melakukan semua proses yang berkaitan dengan aritmatika dan operasi logika yang perlu dilakukan pada kata-kata instruksi

unit pusat pengolahan pusat (*central processing unit*) unit yang melakukan sebagian besar pemrosesan di dalam komputer dengan memproses semua instruksi yang diterima oleh perangkat lunak yang berjalan pada pc dan oleh komponen perangkat keras lainnya, dan bertindak sebagai kalkulator yang kuat

variabel (*variable*) nama simbolik yang digunakan untuk melacak nilai yang dapat berubah saat program berjalan

spasi (*whitespace*) adalah karakter atau rangkaian karakter apa pun yang mewakili spasi horizontal atau vertikal dalam tipografi

wifi (*wifi*) sebuah teknologi yang menggunakan gelombang radio (secara nirkabel) melalui jaringan komputer untuk bertukar data, termasuk koneksi internet yang memiliki kecepatan tinggi

Daftar Pustaka

Aaron. (2019). How the Internet Works in 5 Minutes (video), diakses dari https://www.youtube.com/watch?v=7_LPdttKXPc.

AlgoRythmics. (2011). Bubble-sort with Hungarian ("Csángó") folk dance (video), diakses dari https://www.youtube.com/watch?v=lyZQPjUT5B4

AlgoRythmics. (2011). Insert-sort with Romanian folk dance (video), diakses dari https://www.youtube.com/watch?v=ROalU379l3U

AlgoRythmics. (2011). Merge-sort with Transylvanian-saxon (German) folk dance (video), diakses dari https://www.youtube.com/watch?v=XaqR3G_NVoo

AlgoRythmics. (2011). Quick-sort with Hungarian (Küküllőmenti legényes) folk dance (video), diakses dari https://www.youtube.com/watch?v=ywWBy6J5gz8

Australian Curriculum. (2020, May 20) Computational thinking in the Australian Curriculum: Digital Technologies (video), diakses dari https://www.youtube.com/watch?v=VFcUgSYyRPg

Community Worksho Series. (2019) Handout digital literasi, Dasar Internet, Dasar Search Engine, Dasar Email, Dasar Microsoft Word, Dasar Microsoft Excel, Dasar Microsoft PowerPoint. Diakses melalui http://cws.web.unc.edu/handouts/ pada 18 Desember 2020.

Deitel, P. & Deitel, H. (2016). C: How to Program Edisi ke-8.

Denning, P. J. (2001). The profession of IT

EdGlossary. (2014). The Glossary of Education Reform for Journalists, Parents, and Community Members, diakses dari https://www.edglossary.org/

George, B. (2011). Digital Planet: Tomorrow's Technology and You Edisi 10, Penerbit Pearson.

Graph. (2021). In wikipedia: https://en.wikipedia.org/wiki/Graph_(discrete_mathematics) diakses pada tanggal 20 Januari 2021.

Graph Traversal. (2021). In wikipedia: https://en.wikipedia.org/wiki/Graph_traversal Diakses pada tanggal 1 Januari 2021.

HTTPS. (2021). In Wikipedia: https://id.wikipedia.org/wiki/HTTPS diakses pada tanggal 20 Januari 2021.

Internet sehat dan aman. (2021). In wikipedia: https://id.wikipedia.org/wiki/Internet_Sehat_dan_Aman diakses pada tanggal 20 Januari 2021.

ISTE. (2012, Januari 4) Computational thinking: A digital age skill for everyone (video), diakses dari https://www.youtube.com/watch?v=VFcUgSYyRPg

Jaringan komputer. (2021). In wikipedia: https://id.wikipedia.org/wiki/Jaringan_komputer diakses pada tanggal 20 Januari 2021.

John, S. (n.d). RealPython: Python While Loops (Indefinite Iteration). Diakses dari https://realpython.com/python-while-loop/ pada 22 Desember 2020.

John, S. (n.d). RealPython: Condition Statement in Python. Diakses dari https://realpython.com/python-conditional-statements/ pada 22 Desember 2020.

Jobs ID. (2020) Info Lowongan Kerja Terbaru dan Populer 2020. Diakses dari http://jobs.id pada tanggal 18 November 2020.

Kemdikbud. (n.d). KBBI, Kamus Besar Bahasa Indonesia, diakses dari https://kbbi.kemdikbud.go.id tanggal 28 November 2020

K12CS. (n.d). Computational Thinking. dapat diakses melalui https://k12cs.org/computational-thinking/

Kotsopoulos, D., Floyd L, Khan S., Namukasa I.K, Somanath S., Weber J., Yiu C. (2017). "A Pedagogical Framework for Computational Thinking", DOI 10.1007/s40751-017-0031-2, # Springer International Publishing.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L. (2011). “Computational Thinking for Youth in Practice” by I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, & L. Werner, 2011, ACM Inroads 2 (1): p. 35. Copyright 2011 by the ACM.

Operating System. (2021). In wikipedia: https://en.wikipedia.org/wiki/Operating_system Diakses pada tanggal 20 Desember 2020.

PyData. (n.d). Pandas: Python Data Analysis Library. Diakses dari https://pandas.pydata.org/

Queue. (2021). In wikipedia: https://en.wikipedia.org/wiki/Queue_(abstract_data_type)

Rekursi. (2021). In wikipedia: https://en.wikipedia.org/wiki/Recursion diakses pada tanggal 11 Januari 2021.

Sachs JD. (2012). From millennium development goals to sustainable development goals. The Lancet. 2012 Jun 9;379(9832):2206-11.

Search Algorithm. (2021). In wikipedia: https://en.wikipedia.org/wiki/Search_algorithm diakses pada tanggal 11 Januari 2021.

Stack. (2021): In wikipedia: https://en.wikipedia.org/wiki/Stack_(abstract_data_type) diakses pada tanggal 11 Januari 2021.

Syahrizal, S. (2019 Maret 12). CNBC Indonesia (12 Maret 2019), CBNC Indonesia, diakses melalui https://www.cnbcindonesia.com/tech/20190312195140-37-60263/siapa-minat-ri-butuh-17-juta-tenaga-kerja-it pada 18 Desember 2020.

TOKI. (2020). Learning center Tim Olimpiade Komputer Indonesia (TOKI): http://tlx.toki.id Diakses pada tanggal 25 Desember 2020.

Vic F.-W. (2005). How Computers Work: The CPU and Memory. Diakses melalui https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading04.htm pada 10 November 2020.

Web Scraping. (2021). In wikipedia: https://en.wikipedia.org/wiki/Web_scraping diakses pada tanggal 11 Januari 2021.

Yaroslav, S. (2017). Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems, EMS Surveys in Mathematical Sciences 4 (2):219–320 (2017).

___________. (n.d.). Binary Search. Di akses melalui https://khanacademy.org/computing/computer-science/algorithms/binary-search pada 23 November 2020.

___________. (n.d.). Dataquest. Tutorial: Web Scraping with Python using Beautiful Soup. Diakses dari https://www.dataquest.io/blog/web-scraping-tutorial-python/ pada 18 Desember 2020.

___________. (n.d.). Digital literasi: Diakses dari http://cws.web.unc.edu/ pada tanggal 10 November 2020.

___________. (n.d.). Google Colaboratory. Welcome to Colaboratory. Diakses dari https://colab.research.google.com/.

Sumber Gambar

Gambar 1.5 Pelantikan OSIS (a) Upacara, (b) Nasi Tumpeng untuk acara ramah tamah. Sumber : (a) http://smandatas.sch.id/wp-content/uploads/2016/09/osis3-620x264.jpg, (b) http://asset-a.grid.id/crop/0x0:0x0/x/photo/2018/08/13/3732593769.jpg

Gambar 1.6 Contoh grafik (a) Cluster (kelompok) Data, (b) Layer pada aplikasi Geographical Information System, (c) Infografis menjadi pengguna internet anti-hoax. Sumber: (a) By Chire - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17085333, (b) By Jaknouse at English Wikipedia, CC BY 3.0, https://commons.wikimedia.org/w/ index.php?curid=31727798 (c) https://aptika.kominfo.go.id/wp-content/uploads/2018/11/2108_Perilaku-Pengguna-Internet-Indonesia.png

Gambar 1.7 Komik lucu tentang teknologi. Sumber: https://karangpanas.org/wp-content/uploads/2012/07/8.jpg

Gambar 2.2 Pencarian (a) buku di perpustakaan, (b) informasi di internet. Sumber: (a) Dokumen Kemendikbud, (b) Photo by Anthony Shkraba from Pexels https://images.pexels.com/photos/4348404/pexels-photo-4348404.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500

Gambar 4.3 Contoh *Microcomputer*. Sumber: By Laserlicht - Creative Commons License, https://en.wikipedia.org/wiki/File:Raspberry_Pi_4_Model_B_-_Side.jpg

Gambar 4.4 Contoh *Personal Computer*. Sumber: By Everaldo Coelho, GNU General Public License, https://en.wikipedia.org/wiki/File:Crystal_Project_computer.png

Gambar 4.5 Contoh *Mini PC*, Coffee Lake-U-based Bean Canyon Intel NUC8i5BEK2. Sumber: By Laserlicht - Creative Commons License, https://en.wikipedia.org/wiki/File:Intel_NUC8.jpg

Gambar 4.6 Contoh *Minicomputer* PDP-8e Minicomputer System. Sumber: By Computer History Museum - Public Domain, https://www.computerhistory.org/collections/catalog/102691309

Gambar 4.7 Salah satu contoh Komputer Mainframe IBM z Systems z13. Sumber: By Agiorgio - Creative Commons License, https://en.wikipedia.org/wiki/File:IBM_z13_and_LinuxONE_Rockhopper.jpg.

Gambar 4.8 Sierra/ATS-2 Super Computer. Sumber: By U.S. Department of Energy - Public Domain, https://en.wikipedia.org/wiki/File:Sierra_Supercomputer_(48002385338).jpg

Gambar 5.11 Kabel Data. Sumber: By Fructibus - Public Domain, https://en.wikipedia.org/wiki/File:USB-C_cable_2017_A.jpg

Gambar 5.12 Contoh Ilustrasi Komunikasi Data Menggunakan Bluetooth. Sumber: CCO Public Domain, https://scx1.b-cdn.net/csz/news/ 800a/2018/bluetooth.jpg

Gambar 6.27 Contoh Visualisasi Data Secara *Real-Time*. Sumber: (a) https://cdn.krjogja.com/wp-content/uploads/2020/05/Screen-Shot-2020-05-26-at-14.20.40.png (b) https://cdn-2.tstatic.net/jateng/foto/bank/images/penumpang-tengah-melihat-papan-pengumuman-penerbangan-di-bandara-jenderal-ahmad-yani-semarang.jpg (c) https://cdn-2.tstatic.net/tribunnews/foto/bank/images/kurs-valas-di-bank-mandiri-tbk_20160624_090147.jpg

Gambar 6.28 Visualisasi untuk Data Berat Badan sebagai Alat Memprediksi Pertumbuhan Ideal Bayi. Sumber: https://www.idai.or.id/downloads/WHO/2.BB%20menurut%20usia/Laki-laki/0_6%20bulan.pdf

Gambar 8.2 *Electronic Numerical Integrator and Computer* (ENIAC), komputer elektronik pertama yang dapat diprogram. Sumber: By Unknown author - U.S. Army Photo, Public Domain, https://commons.wikimedia.org/w/index.php?curid=55124

Gambar 8.3 Ada Lovelace, Pemrogram Komputer Paling Awal di Dunia. Sumber: By Alfred Edward Chalon - Science Museum Group, Public Domain, https://commons.wikimedia.org/w/index.php?curid=28131684

Gambar 8.4 Beragam bentuk komputer dalam kehidupan sehari-hari (a) Smart Watch (b) Surface Table, (c) Virtual Reality. sumber: (a) https://commons.wikimedia.org/wiki/File:Smart_watch.jpg (b) https://commons.wikimedia.org/wiki/File:Surface_table.JPG (c) https://en.wikipedia.org/wiki/Virtual_reality#/media/File:Reality_check_ESA384313.jpg,

Gambar 8.8 Tujuan Pembangunan Berkelanjutan. sumber: https://global.unitednations.entermediadb.net/assets/mediadb/services/module/asset/downloads /preset/assets/2015/09/22525/image1170x530cropped.jpg

Gambar 8.9 Ilustrasi Gambaran Dunia di Masa Depan, Masyarakat 5.0 (Society 5.0) (a) abstraksi, (b) contoh aktivitas. Sumber: (a) https://www8.cao.go.jp/cstp/english/society5_0/society5_0e-1.jpg, (b) https://www8.cao.go.jp/cstp/english/society5_0/society5_0e-2.jpg

Gambar 9.2 Contoh Membuat Desain Aplikasi dengan Menggunakan MS PowerPoint. Sumber:https://usabilitygeek.com/wp-content/uploads/2013/02/wireframing-storyboarding-microsoft-powerpoint-powermockup-2.jpg

Indeks

A

AKUMULATOR 86
algoritma 6, 7, 13, 24, 25, 34, 77, 116, 130, 131, 135, 136, 137, 144, 145, 146
Algoritma dan Pemrograman 6, 7, 13, 136
ALU 64, 80, 81, 84
Analisis Data 6, 7, 109, 110, 111, 112, 131
antrean 37, 38, 41, 77
aplikasi 2, 47, 49, 57, 231, 234, 240, 133
aplikasi perkantoran 46, 47, 48, 50, 57
arduino 66, 67
Arithmetic/Logic Unit 80

B

bekerja dalam kelompok 1, 2, 11, 15, 16, 233
berpikir komputasional 3, 5, 7, 25, 78, 137, 139, 187, 191
Bubble Sort 44

C

CD ROM 80
Central Processing Unit 65, 80
coding 137, 148, 162, 190
Computational Thinking 3, 6, 12
Computer Network 6
Computer Science 3
Computing Practices 6
Computing System 6
CPU 64, 65, 66, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89
CU 80, 83

D

Dampak Sosial Informatika 6, 7, 66
Data 6, 7, 11, 20, 34, 57, 58, 84, 86, 87, 99, 100, 101, 102, 103, 104, 110, 111, 112, 113, 115, 116, 120, 121, 122, 125, 126, 127, 128, 129, 130, 131, 132, 134, , 157, 158, 160, 246, 251, 252
Data Analysis 6, 134
debugging 137
Dequeue 43
diagram 47, 48, 49, 51, 53, 54, 78, 79, 127, 128, 129, 130, 132, 137, 138, 139, 140, 141, 142, 143, 144, 146, 147, 169, 176, 177, 179
Diagram Alir 138, 139, 140, 141, 143, 167, 169, 174, 180, 184, 187, 189
Dokumentasi 232

E

EDGE 100
efektif 1, 2, 9, 16, 17, 23, 25, 137
ekspresi 87, 135, 139, 142, 143, 148, 155, 165, 166, 167, 168, 169, 170, 172, 178, 179, 181, 182
Enqueue 43

G

General Packet Radio Service 97, 100
Generic Skill 1, 14
Google 27, 47, 110, 112, 113, 114, 115, 122, 123, 126, 134, , 153, 240
Google Colaboratory 112, 113, 123, 124, 134
GPRS 97, 100, 106
graf 44
Graphical User Interface 69

I

industri 3, 4, 67
infografis 2, 14, 17, 19, 20, 21, 132, 241
informatika 3, 5, 6, 7, 8, 9, 10, 11, 14, 21, 26, 38, 39, 43, 50, 137, 151, 236
Informatika 1, 2, 3, 6, 7, 8, 9, 11, 12, 13, 14, 24, 25, 26, 118, 66, 119, 231, 233, 235, 244
Information and Communication Technology 6
Information System 20, 252
insertion sort 31, 32, 33
Interaksi Manusia dan Komputer 69

Internet 6, 41, 71, 91, 92, 93, 94, 95, 96, 97, 98, 108, 252
internet service provider 92

J

Jaringan Komputer dan Internet 6, 91, 92
jaringan lokal 71, 91, 93, 94, 95, 101, 107

K

keluaran 65, 69, 74, 80, 81, 83, 86, 117, 122, 125, 126, 141, 142, 153, 155, 159, 162, 163, 164, 169, 172, 183, 185, 193, 195
Kolaborasi 15, 70
Koleksi Data 110, 122, 125
komputasi 3, 5, 6, 7, 9, 10, 14, 25, 27, 31, 38, 65, 69, 70, 71, 78, 137, 138, 231
Komputer 6, 7, 13, 24, 61, 63, 64, 65, 66, 67, 68, 69, 70, 73, 76, 79, 80, 81, 83, 87, 91, 92, 98, 99, 105, 106, 157, 193, 194, 65
Komputer Mainframe 68
Komputer Mikro 66
komunikasi 5, 14, 15, 72, 75, 91, 92, 93, 97, 99, 100, 101, 102, 103, 104, 105, 159, 237

M

Mail Merge 46, 57, 62
manajemen proyek 191, 232, 233
masukan 65, 69, 70, 79, 83, 140, 141, 142, 146, 155, 159, 161, 162, 163, 164, 174, 176, 179, 182, 184, 186, 188, 193, 195, 239
masyarakat 4, 14, 65, 180, 233, 234, 235, 236, 239
mesin konseptual sederhana 74, 83, 87, 88
microprocessor 66
Microsoft 46, 47, 48, 49, 50, 59, 149
Microsoft Excel 46, 47, 48, 49, 50
Microsoft PowerPoint 46
Microsoft Word 46, 47, 48, 49, 59
Minicomputer 68
Mini PC 67
multitasking 74, 75, 76, 79

O

Object Linking & Embedding 46, 50, 54, 55
observasi 232, 233, 234, 236, 238, 239, 240, 241
Open Source 134

P

pemrograman 5, 7, 30, 34, 47, 50, 66, 82, 115, 116, 118, 119, 120, 121, 130, 131, 135, 136, 137, 144, 146, 148, 149, 151, 152, 153, 155, 157, 162, 166, 170, 171, 177, 187, 190, 191, 192, 193, 194, 236, 242
Pencarian 26, 27, 44, 74, 113, 114
Pengurutan 30, 31, 44
perangkat keras 3, 5, 6, 7, 10, 11, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 90, 149
perangkat lunak 3, 5, 6, 8, 10, 11, 47, 59, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 90, 136, 151
perencanaan kerja 2, 17
Pop 41, 42, 43
Praktika Lintas Bidang 6, 8, 232, 233
Presentasi 21, 46, 60, 61, 238, 239
problem solving 25, 137
program 5, 7, 9, 14, 16, 26, 30, 31, 65, 66, 70, 71, 72, 73, 74, 75, 76, 80, 81, 82, 83, 84, 85, 101, 109, 113, 114, 116, 117, 118, 120, 121, 122, 123, 124, 125, 127, 128, 130, 131, 132, 133, 135, 136, 137, 138, 140, 143, 146, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 162, 163, 164, 165, 166, 167, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 196, 231, 234, 236, 237, 241, 242
pseudocode 137, 143, 144, 146, 147
Public Switched Telephone Network 95
Push 41, 42

Q

Queue 37, 39, 41, 42

R

RAM 66, 74, 80
raspberry pi 66
Rekursi 44
ROM 80

S

searching 24
sejarah 5, 149
selection sort 31, 34
Sertifikat SSL 92, 105, 106
sistem 3, 5, 6, 7, 9, 10, 14, 25, 63, 64, 65, 66, 68, 69, 70, 71, 72, 73, 74, 76, 78, 79, 90, 93, 97, 131, 132, 147, 149, 151, 152, 180, 187, 231, 235, 236, 237
sistem komputer 3, 10, 25, 63, 64, 66, 69, 70, 71, 78, 97, 131
sistem operasi 7, 63, 64, 65, 66, 68, 69, 70, 71, 72, 73, 74, 76, 90, 149, 151, 152, 180, 187
Social Impact of Informatics 6
software 65, 66, 71, 72
sorting 24, 31
stack 24, 37, 38, 39, 40, 41, 42, 43, 44
Supercomputer 252

T

Teknologi Informasi dan Komunikasi 6, 7, 8, 45, 46
Tumpukan 37, 38

V

variabel 74, 84, 116, 117, 118, 119, 128, 129, 139, 141, 142, 143, 148, 155, 156, 157, 158, 159, 160, 166, 167, 168, 174, 178, 179, 189, 195
Visualisasi Data 110, 126, 127, 128, 129, 130, 132, 252

W

Web Scraping 122, 123, 134

A. Profil Penulis



Informasi Diri

Nama Lengkap : Dr. Mushthofa, SKom, MSc
Email : mush@apps.ipb.ac.id
Instansi : Institut Pertanian Bogor
Alamat Instansi : Kampus IPB Dramaga, Bogor
Bidang Keahlian : Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Dosen/peneliti di Departemen Ilmu Komputer IPB (2009 – sekarang)
- Research Assistant di Ghent University, Belgium (2014 – 2018)

Riwayat Pendidikan dan Tahun Belajar:
- Sarjana Ilmu Komputer, IPB (2005)
- Master's in Computational Logic, TU Vienna (2009)
- Doctor in Computer Science/ Bioinformatics, Ghent University, Belgium (2018)

Judul Buku/Karya dan Tahun Terbit (3 Tahun Terakhir):
- Fuzzy Answer Set Programming: From Theory to Practice (Chapter pada Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications (2020)

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- Modeling multi-valued biological interaction networks using fuzzy answer set programming, 2018, Fuzzy Sets and Systems



Informasi Diri

Nama : Wahyono, S.Kom., Ph.D.
Email : wahyo@ugm.ac.id
Instansi : Universitas Gadjah Mada
Alamat Instansi : Sekip Utara Bulaksumur, Yogyakarta
Bidang Keahlian : Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Staff Pengajar Prodi Ilmu Komputer, UGM, Yogyakarta (2012 – now)
- Sekretaris Program Studi Magister Ilmu Komputer, UGM, Yogyakarta (2021-now)

Riwayat Pendidikan dan Tahun Belajar:
- S3 Teknik Elektro, University of Ulsan, Korea (2012-2017)
- S1 Ilmu Komputer, Universitas Gadjah Mada, Indonesia (2006-2010)

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- Pengembangan Sistem Surveilans Cerdas dan Terintegrasi Berbasis Kamera (2020)



Informasi Diri

Nama : Auzi Asfarian, S.Komp., M.Kom.
Email : asfarian@apps.ipb.ac.id
Instansi : Dept. Ilmu Komputer, FMIPA, Institut Pertanian Bogor
Alamat Instansi : Wing 20, Jl. Meranti Kampus IPB, Babakan, Kec. Dramaga, Bogor, Jawa Barat 16680
Bidang Keahlian : Ilmu Komputer, Interaksi Manusia-Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Dosen, Ilmu Komputer, FMIPA, Institut Pertanian Bogor (2015 – sekarang).
- Vice Chair, Indonesia ACM SIGCHI Chapter (2019 – sekarang).

Riwayat Pendidikan dan Tahun Belajar:
- Magister Ilmu Komputer, Institut Pertanian Bogor (2012 – 2014)
- Sarjana Ilmu Komputer, Institut Pertanian Bogor (2007 – 2012)

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- E-Initiative for Food Security: Design of Mobile Crowdfunding Platform to Reduce Food Insecurity in Indonesia Using The Wheel Methods – 2020

Informasi Diri



Nama Lengkap : Dean Apriana Ramadhan, SKomp, MKom
Email : deanaprianaramadhan@apps.ipb.ac.id
Instansi : Departemen Ilmu Komputer, FMIPA IPB
Alamat Instansi : Departemen Ilmu Komputer Jl Meranti Wing 20, Level 5 Kampus IPB Darmaga 16680
Bidang Keahlian : Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Dosen Departemen Ilmu Komputer IPB
- Direktorat Sistem Informasi dan Transformasi Digital, IPB University

Riwayat Pendidikan dan Tahun Belajar:
- Sarjana Ilmu Komputer, Departemen Ilmu Komputer IPB. Lulus 2012
- Magister Ilmu Komputer, Departemen Ilmu Komputer IPB. Lulus 2015

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- Ramadhan DA, Nurhadryani Y. Hermadi I. 2014. Campaign 2.0: Analysis Of Social Media Utilization In 2014 Jakarta Legislative Election. International Conference on Advanced Computer Science and Information Systems (JCACSIS} 2014. Jakarta (ID): Fasilkom UI.

Informasi Diri



Nama Lengkap : Hanson Prihantoro Putro
Email : hanson @uii.ac.id
Instansi : Universitas Islam Indonesia
Alamat Instansi : Jl Kaliurang Km 14,5 Sleman Yogyakarta
Bidang Keahlian : Informatika / Rekayasa Perangkat Lunak

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Dosen Informatika, Universitas Islam Indonesia (2012 - sekarang)
- Programmer, PT Lapi Divusi Bandung (2009 - 2011)

Riwayat Pendidikan dan Tahun Belajar:
- S2 Informatika, Institut Teknologi Bandung (2009 - 2011)
- S1 Informatika, Institut Teknologi Bandung (2005 - 2009)

Judul Buku/Karya dan Tahun Terbit (3 Tahun Terakhir):
- Diktat Praktikum Pemrograman Berorientasi Obyek, Laboratorium Komputasi dan Sistem Cerdas, Jurusan Teknik Informatika, Universitas Islam Indonesia (2012).

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- Pengembangan Sistem Informasi Akademik MI-Gateway Berbasis Website, Kolokium Automata (2019).

Informasi Diri



Nama : Irya Wisnubhadra
Email : irya.wisnubhadra@uajy.ac.id
Instansi : Universitas Atma Jaya Yogyakarta
Alamat Instansi : Jl. Babarsari 44, Yogyakarta
Bidang Keahlian : Pemrograman, Database System, Business Intelligence

Data Penelitian dan Karya detail dapat dilihat di Google Scholar

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Dosen Pengajar Tetap, Teknik Informatika Universitas Atma Jaya Yogyakarta

Riwayat Pendidikan dan Tahun Belajar:
- S1: Department Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada
- S2: Teknik Informatika, Rekayasa Perangkat Lunak, Institut Teknologi Bandung
- S3: Faculty of ICT, Universiti Teknikal Malaysia, Melaka (2018-sekarang)

Judul Buku/Karya dan Tahun Terbit (3 Tahun Terakhir):
- Copyright Perangkat Lunak, Aplikasi monitoring transportasi buah sawit, logtransawit.online, 2019

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- Pengembangan mobility business intelligence untuk peningkatan produktivitas sistem transportasi TBS kelapa sawit secara berkelanjutan, Penelitian Terapan, Tahun 2020 – 2021, DIKTI

Informasi Diri



Nama Lengkap : Heni Pratiwi
Email : heni.pr@gmail.com
Instansi : SMP Negeri 2 Yogyakarta
Alamat Instansi : Jl. P. Senopati No. 28-30 Yogyakarta
Bidang Keahlian : Mengajar Informatika / TIK

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):
- Tahun 2009 - Sekarang; Guru TIK SMP
- Tahun 2010 - Sekarang; Pengurus MGMP TIK Kota Yogyakarta

Riwayat Pendidikan dan Tahun Belajar:
- Universitas Sanata Dharma (2007)
- Universitas Negeri Yogyakarta (2009)

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):
- Penerapan Model Pembelajaran Card Sort Untuk Meningkatkan Motivasi Belajar TIK di Kelas IX C SMP Negeri 2 Yogyakarta Semester 1 Tahun Pelajaran 2017/2018 (2017)

Informasi Diri



Nama Lengkap : Budiman Saputra
Email : suratbudiman@gmail.com
Instansi : SMPN 6 Yogyakarta
Alamat Instansi : R.E.Monginsidi No.1 Yogyakarta
Bidang Keahlian : Tehnik Informatika

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir) :
- Guru SMPN 6 Yogyakarta (sekarang)

Riwayat Pendidikan dan Tahun Belajar :
- D3 STIMIK AKAKOM 1997
- S1 UPY 2007
- S2 Universitas Widya Wiwaha 2018

B. Profil Penelaah



Informasi Diri

Nama lengkap : Dr. Inggriani
Email : inge@informatika.org
Instansi : Bebras Indonesia, ITB, IT Del
Bidang Keahlian : Informatika

Data Penelitian dan Karya detail dapat dilihat di Google Scholar

Riwayat Pekerjaan (3 Tahun Terakhir):

- Anggota Asesor BAN PT (2014-sekarang)
- Anggota Senat Akademik Institut Teknologi Del (2014-sekarang)
- Dosen STEI ITB (1977-2018) – purnabakti

Riwayat Pendidikan Tinggi dan Tahun Belajar :

- 1977: Bachelor of Engineering Physics.
- 1985: Master DESS-IDC (Diplôme D'Etudes Supérieures Spécialisées, Informatique Double Compétence), Université Grenoble I, France.
- 1986: Master DEA Informatique, Institute Nationale Politechnique de Grenoble, France.
- 1989: Doctor en Informatique, Université Joseph Fourier, Grenoble, France.

Judul Buku/Karya Yang Pernah Ditulis dan Tahun Terbit (5 tahun terakhir):

- Rouvrais S., Chelin N., Gerwel P. C., Audunsson H., Liem Inggriani., Tudela V. L., "Preparing 5.0 Engineering Students for an Unpredictable Post-COVID World", World Engineering Education Forum and the Global Engineering Deans Council (WEEF/GEDC) Virtual Conference, 16 – 19 November 2020.

Informasi Diri



Nama Lengkap : Paulina Heruningsih Prima Rosa, S.Si., M.Sc.
Email : rosa@usd.ac.id
Instansi : Universitas Sanata Dharma (USD)
Alamat Instansi : Kampus III, Paingan, Maguwoharjo, Depok Sleman, Yogyakarta 55282
Bidang Keahlian : Informatika / Ilmu Komputer

Data Penelitian dan Karya detail dapat dilihat di Google Scholar

Riwayat Pekerjaan/Profesi

- Dosen Prodi Teknik Informatika USD : 2008 – sekarang
- Wakil Dekan I FST USD : 2015 - 2019

Riwayat Pendidikan dan Tahun Belajar:

- 1988 -1993: S1 Prodi Ilmu Komputer - Universitas Gadjah Mada
- 1996 -1999: S2 Department of Computer Science - Ateneo de Manila University, Philippines

Judul Buku/Karya dan Tahun Terbit (3 Tahun Terakhir):

- Kontributor artikel dalam Buku Kumpulan Hasil Penelitian Tentang Pemilu, Penerbit Universitas Sanata Dharma, 2015.

Judul Penelitian dan Tahun Terbit (3 Tahun Terakhir):

- P.H.P Rosa, H. Sriwindono, R.A. Nugroho, K. Pinaryanto, 2020, Comparison of Crossover and Mutation Operators to Solve Teachers Placement Problem by Using Genetic Algorithm, Journal of Physics: Conference Series, Vol. 1566, July 2020

Informasi Diri



Nama lengkap : Adi Mulyanto
Email : adi@informatika.org
Instansi : Institut Teknologi Bandung
Alamat Kantor : Jl. Ganesha 10 Bandung
Bidang Keahlian : Informatika

Data Penelitian dan Karya detail dapat dilihat di Google Scholar

Riwayat Pekerjaan (10 Tahun Terakhir):

- Dosen Informatika Institut Teknologi Bandung (1997 – sekarang)
- Konsultan Teknologi Informasi (1994 – sekarang)

Riwayat Pendidikan Tinggi dan Tahun Belajar :

- Sarjana Teknik Informatika ITB – Lulus 1994
- Magister Informatika ITB – Lulus 1997

Judul Buku/Karya Yang Pernah Ditulis dan Tahun Terbit (3 tahun terakhir):

- Belajar Pemrograman Secara On Line dan Jarak Jauh, Pengenalan Sistem Penilaian Program Secara Otomatis Untuk Indonesia. Tahun 2015.

Judul Penelitian dan Tahun Terbit (3 tahun terakhir):

- Repositori Infomasi Objek Wisata dengan Teknologi Semantik Web dan Basis data Multimedia untuk Pengelolaan dan Promosi Desa Wisata. Tahun 2012 s.d 2014.

B. Profil Editor

Informasi Diri



Nama Lengkap : Dr.rer.nat. Cecilia Esti Nugraheni, S.T., M.T.
E-mail : cheni@unpar.ac.id
Alamat Kantor : Universitas Katolik Parahyangan,
Jl. Ciumbuleuit 94 Bandung 40141
Bidang keahlian : Teknik Informatika

Riwayat pekerjaan/profesi dalam 3 tahun terakhir:

- Dosen Tetap Program Studi Teknik Informatika Universitas Katolik Parahyangan.

Riwayat Pendidikan Tinggi dan Tahun Belajar:

- S3: Institut für Informatik, Ludwig Maximilians Universität (Univ. of Munich), Germany (1999-2004).
- S2: Teknik Informatika, Institut Teknologi Bandung (1995-1997).
- S1: Teknik Informatika, Institut Teknologi Bandung (1988-1993).

Publikasi (10 tahun terakhir):

- C. E. Nugraheni, L. Abednego, M.Widyarini. A Combination of Palmer Algorithm and Gupta Algorithm for Scheduling Problem in Apparel Industry. IJFLS. Vol. 11 no. 1, 2021.

Judul penelitian dan Tahun Terbit (3 tahun terakhir):

- Pengembangan Perangkat Lunak Penjadwalan Proses Produksi Pakaian Jadi Pada Usaha Kecil Menengah dengan Pendekatan Hiperheuristik. Hibah Penelitian Terapan Unggulan Perguruan Tinggi Ristekdikti. 2018 – 2020.



Informasi Diri

Nama : Dr. Christina Tulalessy
Kantor : Pusat Kurikulum dan Perbukuan
Email Penulis : nonatula6@gmail.com
Akun Media Sosial : christina tulalessy
Bidang Keahlian : Kurikulum, Penelitian dan Evaluasi
Pendidikan, Editor

Riwayat pekerjaan/profesi dalam 10 tahun terakhir:
- Pusat Perbukuan 1988–2010
- Pusat Kurikulum dan Perbukuan 2010–saat ini
- Asesor Kompetensi Penulis dan Penyunting BNSP

Riwayat Pendidikan Tinggi dan Tahun Belajar:
- S3 Penelitian dan Evaluasi Pendidikan UNJ 2017
- S2 Penelitian dan Evaluasi Pendidikan UHAMKA 2006
- S1 Tata Busana IKIP Jakarta 1988

Publikasi (10 tahun terakhir):
1. Penelitian Tindakan Kelas: Apa, Mengapa, Bagaimana: 2020

D. Profil Ilustrator dan Desainer

Informasi Diri

Nama Lengkap : Malikul Falah
Surel : malikulfalah@gmail.com
Alamat Kantor : Grand Kahuripan Blok EG 16 Klapanunggal Bogor
Bidang Keahlian : Desain Grafis/Ilustrasi

Riwayat Pekerjaan:
1. 2015-sekarang : Konsultan desain dan penerbitan (Bogor)
2. 2013-2015 : Desainer PT Zaytuna Ufuk Abadi (Jakarta)

Riwayat Pendidikan Tinggi dan Tahun Belajar:
1. 2008-2009 : Fakultas Ekonomi Universitas Pancasila
2. 2001-2002 : Diploma Desain Grafis PIKMI Garuda Bandung

Buku yang Pernah dibuat Ilustrasi dan Tahun Pelaksanaan (10 Tahun Terakhir):
- Rahasia di Balik Penciptaan Organ Tubuh Manusia. Zaytuna Ufuk Abadi 2017

Penghargaan (10 Tahun Terakhir):
- Penulis terpilih Gerakan Literasi Nasional Badan Pengembangan dan Pembinaan Bahasa, Kemendikbud 2019.
- Desainer Terbaik Kategori Fiksi Remaja, Indonesia Membumi KPK-IKAPI 2015.

Informasi Diri

Nama Lengkap : Nurul Fatonah
Surel : nurul.fatonah@gmail.com
Bidang Keahlian : Desain dan Editor

Riwayat Pekerjaan:
- Editor Kubus Media Group (2017-2018)
- Desainer Buku Pendidikan Agama Kristen Kelas VIII dan XI, Pendidikan Agama Buddha Kelas III, Pendidikan Agama Islam Kelas V Kurikulum 2013 (2013-2018)

Riwayat Pendidikan Tinggi dan Tahun Belajar:
- D3: Teknik Grafika dan Penerbitan, Politeknik Negeri Jakarta. 2011